

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
19 April 2001 (19.04.2001)

PCT

(10) International Publication Number
WO 01/27874 A1

(51) International Patent Classification?: G06N 3/02

(21) International Application Number: PCT/US00/28453

(22) International Filing Date: 13 October 2000 (13.10.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
09/418,099 14 October 1999 (14.10.1999) US

(63) Related by continuation (CON) or continuation-in-part (CIP) to earlier application:
US 09/418,099 (CIP)
Filed on Not furnished

(71) Applicants (for all designated States except US): THE
SALK INSTITUTE [US/US]; 10010 North Torrey Pines

Road, La Jolla, CA 92037 (US). CARNEGIE-MELLON
UNIVERSITY [US/US]; 4615 Forbes Avenue, Suite 302,
Pittsburgh, PA 15213 (US).

(72) Inventors; and

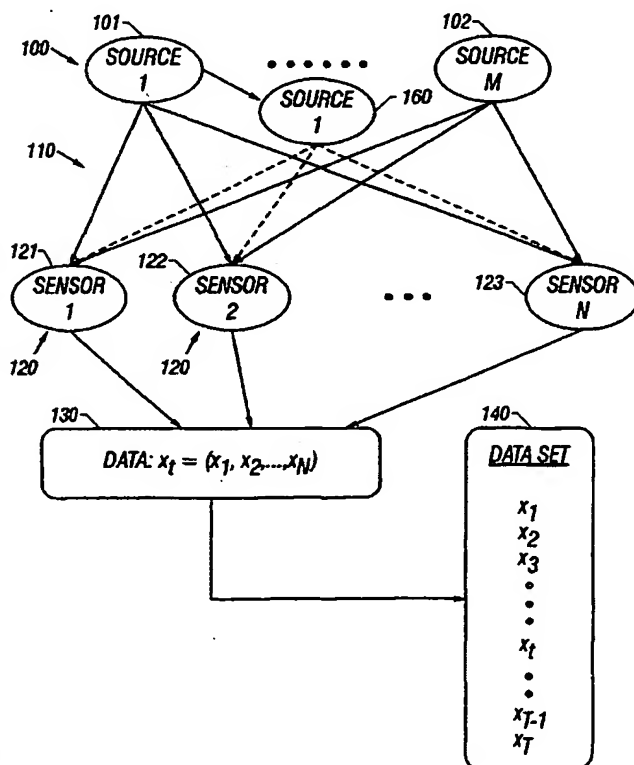
(75) Inventors/Applicants (for US only): LEE, Te-Won
[DE/US]; 8328 Regents Road #2F, San Diego, CA 92122
(US). LEWICKI, Michael [US/US]; 555 South Megley
Avenue #3, Pittsburgh, PA 15232 (US). SEJNOWSKI,
Terrance, J. [US/US]; 672 San Marino Drive, Solana
Beach, CA 92075 (US).

(74) Agent: MCFARLAND, James, D.; Suite 280A, 12555
High Bluff Drive, San Diego, CA 92130 (US).

(81) Designated States (national): AE, AG, AL, AM, AT, AU,
AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ,
DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR,
HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR,
LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ,

[Continued on next page]

(54) Title: UNSUPERVISED ADAPTATION AND CLASSIFICATION OF MULTI-SOURCE DATA USING A GENERALIZED GAUSSIAN MIXTURE MODEL



(57) Abstract: A computer-implemented method and apparatus that adapts class parameters, classifies data and separates sources configured in one of multiple classes whose parameters are initially unknown. The data set may be generated in a dynamic environment where the sources (100) provide signals are mixed and received by sensors (120), and the mixing parameters change without notice and in an unknown manner. A generalized Gaussian mixture model is used to classify the observed data into two or more mutually exclusive classes whose basis functions may be defined by a range of probability density functions. The class parameters for each of the classes are adapted to a data set in an adaptation algorithm in which class parameters including mixing matrices, bias vectors, and pdf parameters are adapted. Each data vector is assigned to one of the learned mutually exclusive classes. In some embodiments the class parameters may have been previously learned, and the system is used to classify the data and if desired to separate the sources. The adaptation and classification algorithms can be utilized in a wide variety of applications such as speech processing, image processing, medical data processing, satellite data processing, antenna array reception, and information retrieval systems.

WO 01/27874 A1



NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM,
TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

Published:

— With international search report.

- (84) Designated States (*regional*): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

UNSUPERVISED ADAPTATION AND CLASSIFICATION OF MULTI-SOURCE DATA USING A GENERALIZED GAUSSIAN MIXTURE MODEL

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention generally relates to computer-implemented systems for processing data that includes one or more source signals, and particularly to systems for
5 adapting parameters to the data, classifying the data, separating sources, or encoding the data.

2. Description of Related Art

Recently, techniques for source separation and efficient data encoding using ICA (Independent Component Analysis) have received attention because of their potential signal processing applications, such as speech enhancement, image processing, telecommunications,
10 and medical signal processing, among others. ICA is a technique for finding a linear non-orthogonal coordinate system in multivariate data. The directions of the axes of the coordinate system are determined by the data's second- and higher-order statistics. The separation is "blind" because the source signals are observed only as unknown linear mixtures of signals from multiple sensors, and the characteristic parameters of the source signals are unknown
15 except that the sources are assumed to be independent. In other words, both the source signals and the way the signals are mixed is unknown. The goal of ICA is to use the data to learn the statistical parameters and basis functions that characterize the sources, and recover the independent sources (i.e., separate the independent sources) given only the unknown linear mixtures of the independent source signals as observed by the sensors. In contrast to
20 correlation-based transformations such as principal component analysis (PCA), the ICA technique adapts a matrix to linearly transform the data and reduce the statistical dependencies of the source signals, attempting to make the source signals as independent as possible. ICA has proven a useful tool for finding structure in data, and has been successfully applied to processing real world data, including separating mixed speech signals and removing artifacts
25 from EEG recordings.

U.S. Patent 5,706,402, entitled "Blind Signal Processing System Employing Information Maximization to Recover Unknown Signals Through Unsupervised Minimization of Output Redundancy", issued to Bell et al. on January 6, 1998, discloses an unsupervised learning algorithm based on entropy maximization in a single-layer feedforward neural network. In the
30 ICA algorithm disclosed by Bell, an unsupervised learning procedure is used to solve the blind signal processing problem by maximizing joint output entropy through gradient ascent to minimize mutual information in the outputs. In that learned process, a plurality of scaling weights and bias weights are repeatedly adjusted to generate scaling and bias terms that are used to separate the sources. The algorithm disclosed by Bell is limited to a single statistical

model for all the sources; i.e. it models all of the sources with only a single probability density function (pdf). Particularly, Bell models all of the sources with a super-Gaussian distribution, which can be described as sharply peaked with heavy tails. Bell does not disclose how to separate sources that have sub-Gaussian pdfs, nor how to separate sources that have pdfs
5 varying from the assumed pdf, such as mixed sub-Gaussian and super-Gaussian source pdfs. Using the algorithm disclosed by Bell, if the pdf assumption is incorrect, (e.g. if one of the sources is sub-Gaussian) source separation can become difficult or impossible.

In pattern classification techniques such as ICA, performance is often determined by how well the underlying statistical distribution of each of the sources is modeled by the
10 classification technique. In a simple case if the source distributions are all assumed to be Gaussian, then the ICA technique is approximately equivalent to principal component analysis (PCA); i.e., PCA assumes the data to be distributed according to a multi-variant Gaussian. However; in many real world pattern recognition problems there may be a number of data clusters whose underlying pdfs are substantially different; for example one data cluster may
15 have a Gaussian pdf, another may have a super-Gaussian pdf, and still another may have a sub-Gaussian pdf. In other words, in most pattern recognition problems there may be a number of different data clusters in which each cluster can be better fitted to a non-Gaussian distribution to model the ensemble of data classes. Standard ICA techniques such as that disclosed by Bell can be used to model non-Gaussian sources (including sub-or super-
20 Gaussian probability density functions); however only a single pdf is assumed for all of the sources. If the assumed pdf is incorrect, then the accuracy of the technique is reduced.

Another problem with ICA relates to the time-varying nature of the data. In many real world situations the conventional ICA algorithm cannot be effectively used because conventional ICA requires the sources to be independent (e.g. stationary), which means that
25 the mixture parameters must be identical throughout the entire data set. If the sources become non-stationary at some point then the mixture parameters will change and the ICA algorithm will not operate properly. For example, in the classic cocktail party example where there are several voice sources, ICA will not operate if one of the sources has moved at some time during data collection because the source's movement changes the mixing parameters. In
30 summary, the ICA requirement that the sources be stationary greatly limits the usefulness of the ICA algorithm to find structure in data.

It would be an advantage to provide a system that generalizes the mixture model to a wide variety of pdfs and a continuously varying form, which can provide greater accuracy and better performance overall.

35

SUMMARY OF THE INVENTION

A mixture model is implemented in which the observed data is categorized into two or more mutually exclusive classes, each class being modeled with a mixture of independent components. The multiple class model allows the sources to become non-stationary. A

computer-implemented method and apparatus is disclosed that adapts multiple class parameters in an adaptation algorithm for a plurality of classes whose parameters (i.e. characteristics) are initially unknown. In the adaptation algorithm, an iterative process is used to define multiple classes for a data set, each class having a set of mixing parameters including a mixing matrix A_k and a bias vector b_k . After the adaptation algorithm has completed operations, the class parameters and the class probabilities for each data vector are known, and data is then assigned to one of the learned mutually exclusive classes. The sources can now be separated using the source vectors calculated during the adaptation algorithm. Advantageously, the sources are not required to be stationary throughout the data set, and therefore the system can classify data in a dynamic environment where the mixing parameters change without notice and in an unknown manner. The system can be used in a wide variety of applications such as speech processing, image processing, medical data processing, satellite data processing, antenna array reception, and information retrieval systems. Furthermore, the adaptation algorithm described herein is implemented in one embodiment to model the data using a wide range of pdfs as defined by generalized Gaussian pdfs, which provides a way to adapt and classify data, and separate sources that have a wide variety of statistical structures.

A computer-implemented method is described that adapts class parameters for a plurality of classes and classifies a plurality of data vectors having N elements that represent a linear mixture of source signals into said classes. The method includes receiving a plurality of data vectors from data index $t = 1$ to $t = T$, initializing parameters for each class, including the number of classes, the probability that a random data vector will be in class k , the mixing matrix for each class, and the bias vector for each class. In a main adaptation loop, for each data vector from data index $t = 1$ to $t = T$, steps are performed to adapt the class parameters, including the mixing matrices and bias vectors for each class. The main adaptation loop is repeated a plurality of iterations while observing a learning rate at each subsequent iteration, and after observing convergence of said learning rate, then assigning each data vector to one of said classes. The source vectors, which are calculated for each data vector and each class, can then be used to separate source signals in each of said classes.

In order to provide a way to separate sources having different pdfs (e.g. sub-Gaussian and super-Gaussian pdfs), an extended infomax ICA learning rule can be used. The extended infomax learning rule associates each source (each basis function) with one of two predefined functions, which generally include one super-Gaussian function and one sub-Gaussian function. In the learning process using the extended infomax rule, the basis functions switch between the two functions until the best fit is found. By using the extended infomax rule and fitting the data to either the sub-Gaussian pdf or the super-Gaussian pdf, the sources can be separated with reasonable accuracy, which is adequate for some uses of ICA. However, many ICA uses, such as data encoding require that each of the sources be more accurately modeled than would be possible using only two pdfs.

In order to improve accuracy over the extended infomax model in the presence of multiple source pdfs, in one embodiment, the mixing matrices are adapted using a generalized Gaussian mixture model to provide a continuously-defined parametric form of the underlying density of each basis function for each class. An exponential power distribution is used to model distributions that deviate from the normal Gaussian distribution, and to provide a general method for modeling non-Gaussian statistical structures. By learning the random variable β , as described herein, a wide class of statistical distributions can be modeled including uniform, Gaussian, Laplacian, and other sub-and super-Gaussian densities. A different pdf can be used for each basis function, for each class. The formulation of a mixture model in the generalized Gaussian model includes as a special case the standard Gaussian mixture model when all sources are Gaussian. The generalized Gaussian mixture model is used to infer the degree of non-Gaussian statistical structure for one or more classes of multi-dimensional densities. This system can be applied to situations where multiple classes exist with unknown source densities. This model can substantially improve classification accuracy compared with standard Gaussian mixture models and can accurately model structure in multi-dimensional data.

A method is also described in which a plurality of data vectors are classified using previously adapted class parameters. The class probability for each class is calculated and each data vector is assigned to one of the previously adapted class. This classification algorithm can be used, for example to compress images or to search an image for a particular structure or particular types of structure.

The method can be used in a variety of signal processing applications to find structure in data, such as image processing, speech recognition, and medical data processing. Other uses include image compression, speech compression, and classification of images, speech, and sound.

BRIEF DESCRIPTION OF THE DRAWINGS

For a more complete understanding of this invention, reference is now made to the following detailed description of the embodiments as illustrated in the accompanying drawing, wherein:

Fig. 1 is a diagram that shows a plurality of M sources that generate signals, a plurality of N sensors that receive mixed signal, a data vector whose elements are defined by the mixed signals from the sensors, and a data set defined by a collection of data vectors;

Fig. 2 is a flow chart of an unsupervised adaptation and classification algorithm that adapts class parameters, classifies the data, and separates the sources;

Fig. 3 is a flow chart of the main adaptation loop shown in Fig. 2;

Fig. 4 is flow chart of the initial calculation loop shown in Fig. 3

Fig. 5 is flow chart of the mixing matrix adaptation loop shown in Fig. 3

Fig. 6A is flow chart of the bias vector adaptation loop shown in Fig. 3;

Fig. 6B is flow chart of the pdf vector adaptation loop shown in Fig. 3;

Fig. 7 is flow chart of operations in the step to adapt number of classes shown in Fig. 2;

Fig. 8 is a graph that shows the results of an experiment to adapt and classify two-dimensional data;

5 Fig. 9A is a graph of data collected over time from a first channel;

Fig. 9B is a graph of data collected over time from a second channel;

Fig. 9C is a graph of a first source (voices) after adapting the parameters, classifying the source vectors, and separating the sources;

10 Fig. 9D is a graph of a second source (background music) after adapting the parameters, classifying the source vectors, and separating the sources;

Fig. 9E is a graph of the class probability for single samples;

Fig. 9F is a graph of the class probability for samples in blocks of 100 adjacent samples;

Fig. 9G is a graph of the class probability for samples in blocks of 2000 adjacent samples;

15 Fig. 10 is a diagram illustrating a variety of source data, a computer to process the data, and output devices;

Fig. 11 is a flow chart of an adaptation (training) algorithm that learns the class parameters based upon a selected data set;

20 Fig. 12 is a flow chart of a classification algorithm that utilizes previously-adapted class parameters to classify a data set;

Fig. 13 is a diagram of an image, illustrating selection of patches and pixels within the patches that are used to construct a vector;

Fig. 14 is a diagram of four image regions, each region having different features that are used to adapt the class parameters for four classes;

25 Fig. 15 is a graph of the number of source vectors as a function of their value, illustrating that values of the source vectors are clustered around zero;

Fig. 16 is a diagram of data collection from a single person and a single microphone;

Fig. 17A shows the generalized Gaussian (exponential power) distribution for $\beta = -0.75$, which is approximately a uniform distribution;

30 Fig. 17B is a graph showing the generalized Gaussian pdf for $\beta = -0.25$;

Fig. 17C is a graph showing the generalized Gaussian distribution for $\beta = 0$, which corresponds to a normal Gaussian distribution;

Fig. 17D is a graph showing the generalized Gaussian pdf for $\beta = 0.5$, which approximates an ICA tanh function;

35 Fig. 17E is a graph showing the generalized Gaussian pdf for $\beta = 1$, which is the Laplacian pdf;

Fig. 17F is a graph showing the generalized Gaussian pdf for $\beta = 2$, which is a narrow, pointed distribution;

Fig. 18A is a scatter plot that shows a two-dimensional data distribution;

Fig. 18B is a histogram of the distribution of the coefficients along the first axis in Fig.

18A;

Fig. 18C is a histogram showing distribution of the coefficients along a second axis in

5 Fig. 18A;

Fig. 18D is a graph of the learned values of the pdf parameter β along the first axis;

Fig. 18E is a graph showing the learned distribution along the second axis of Fig. 18A;

and

10 Fig. 19 is a graph showing learned results using simulated data in which four classes were simulated and then found using the generalized Gaussian mixture model as described herein.

DETAILED DESCRIPTION

This invention is described in the following description with reference to the Figures, in which like numbers represent the same or similar elements.

15 The following symbols are used herein to represent the certain quantities and variables, and in accordance with conventional usage, a matrix is represented by an uppercase letter with boldface type, and a vector is represented by a lowercase letter with boldface type.

Table of Symbols

A_k	mixing matrix with elements a_{ij} for class k
A^{-1}	filter matrix, inverse of A
b_k	bias vector for class k
β	pdf parameter that specifies a particular generalized Gaussian pdf
C	designation of class: e.g. class 1, class 2, etc.
β_k	vector of pdf parameters β for basis functions of class k
θ_k	parameters for class k
Θ	parameters for all classes
g	random variable exponent in generalized Gaussian model
J	Jacobian matrix
k	class index
K	number of classes
μ	mean of generalized Gaussian pdf
q_k	switching moment vectors for sub- and super-Gaussian densities
Q_k	diagonal matrix with elements of the vector q_k
M	number of sources
n	mixture index
N	number of sensors (mixtures)
$p(s)$	probability density function

- 7 -

s_i	Independent source signal vectors
σ	standard deviation of generalized Gaussian pdf
t	data index, (e.g. time or position)
T	total number of data vectors in the data set
W	weight matrix
x_t	observed data vector (data point) at data index t
X	observed data vectors $X = [x_1, \dots, x_t, \dots, x_T]^T$ (whole data set)

In some instances, reference may be made to "basis functions" or "basis vectors", which are defined by the columns of the mixing matrix. In other words, the basis functions or vectors for a class are defined by the column vectors of the mixing matrix for that class.

5 Overview of a Data Set

Reference is now made to Fig. 1 which shows a plurality of M sources 100, including a first source 101, a second (M th) source 102, and a number of sources in-between. The sources 100 provide signals shown generally at 110 to a plurality of N sensors 120, including a first sensor 121, a second sensor 122, a third (N th) sensor 123, and a number of sensors in-between that depend upon the embodiment. From Fig. 1 it can be seen that the sensors receive a linear combination (mixture) of the signals from the sources. The number of sensors (N) is assumed to be greater than or equal to the number of sources (M), i.e. $N \geq M$. Subject to this restriction, there is no upper limit on the number of sources M and sensors N , and accordingly M and N are constrained only by practical concerns.

The actual number of sources may be unknown, and in such circumstances it may be useful to estimate the number of sources. If the number of sensors is greater than or equal to the number of sources, then the ICA algorithm will work in the adaptation process described herein. However if the number of sensors is less than the number of sources, then an alternative to ICA must be used. One way of estimating the number of sources is to compute the correlation matrix of the data set X . The rank of the correlation matrix gives an estimate of the number of actual sources in the data.

The parameters (e.g. characteristics) of the mixture and the sources are initially unknown. The sources 100 are assumed to be mutually independent, and each of their probability distributions is assumed to be non-Gaussian. The sources and sensors may comprise many different combinations and types. For example, each of the sources may be a person speaking in a room, in which case the signals comprise voices provided to N microphone sensors situated in different locations around the room. All the voices are received by each microphone in the room, and accordingly each microphone outputs a linear combination (a mixture) of all the voices. The data from each of the microphones is collected in a data vector x_t shown at 130 that has N elements, each element representing data from its corresponding sensor. In other words the first element x_{t1} includes data from the first sensor,

the second element x_2 includes data from the second sensor, and so forth. In the microphone example, the data vectors may be collected as a series of digital samples at a rate (e.g. 8 kHz) sufficient to recover the sources. In some embodiments it may be convenient to process blocks of data instead of a single vector; e.g., it may be useful to combine the data vectors from x_1 to x_{100} , and treat the combined vectors, for purposes of performing adaptation and/or classification, as a single block of data.

A series of observations of the sources are observed by the sensors from $t = 1$ to $t = T$. Typically the variable t represents time, and accordingly the series of measurements typically represent a time sequence of observations. The observed data vectors are collected in a data set 140, which includes a group of all observed data vectors from x_1 to x_T . The data log may reside in the memory of a computer, or any other suitable memory location from which it can be supplied to a computer for processing. Before processing, the data vectors must be in digital form, and therefore if the information from the sensors is not already in digital form, the data must be digitized by any suitable system. For example if the microphones receive analog signals, these signals must be processed by an audio digitizer to put the data in a digital form that can be stored in a computer memory and processed.

Separation of Sources

Based upon the mixed signals received by the sensors 120, one goal in some embodiments is to separate the sources so that each source can be observed. In the above example, this means that the goal is to separate the voices so that each voice can be listened to separately. In other embodiments to be described, the data set may include patches from digitized images in which the N elements include data from N pixels, or even data from a single sensor such as a microphone in which the N elements include a series of N samples over time.

If the sources are independent for all observations from $t = 1$ to T , then an ICA (Independent Components Analysis) algorithm such as disclosed by Bell in U.S. Patent 5,706,402, which is incorporated by reference herein, can be utilized to separate the sources. In the ICA algorithm disclosed by Bell, an unsupervised learning procedure is used to solve the blind signal processing problem by maximizing joint output entropy through gradient ascent to minimize mutual information in the outputs. In that learned process, a plurality of scaling weights and bias weights are repeatedly adjusted to generate scaling and bias terms that are used to separate the sources. However, the ICA algorithm disclosed by Bell is limited because the sources must be independent throughout the data set; i.e. Bell's ICA algorithm requires that the sources must be independent for all data vectors in the data log. Therefore, if one of the sources becomes dependent upon the other, or in the example above if one of the sources shifts location, such as the first sensor 101 moves to the location shown in dotted lines at 160, the mixture parameters for the signals 110 will change and Bell's ICA algorithm will not operate properly.

The algorithm described herein provides a way to classify the data vectors into one of multiple classes, thereby eliminating the assumption of source independence throughout the data set, and allowing for movements of sources and other dependencies across data vectors. However, the sources in each data vector are still assumed to be independent.

5 Class Characteristics (Parameters)

Each class has a plurality of different parameters in the form of a mixing matrix A_k , a bias vector b_k , and a class probability $p(C_k)$. In addition, the class parameters may include density function parameters indicative of the pdf for each of basis functions (e.g. sources) in the mixing matrix. However, because the parameters for each class are initially unknown, one goal is to determine the class characteristics (i.e. determine the parameters). The algorithm described herein learns the parameters for each class in a process that includes adapting (i.e. learning) the mixing matrix and bias vectors in an iterative process. Optionally, the class probability can also be adapted. Once adapted, each data vector is assigned to a mutually exclusive class, the corresponding source vector calculated for the assigned class is used as the desired source vector.

The characteristic parameters for each class are referenced by the variable θ_k from $k = 1$ to K . Each class has a probability designated by $p(C_k)$, which is the probability that a random data vector will fall within the class k . The characteristics for all K classes are collectively referenced by Θ . The description of the parameters for each class may vary between embodiments, but typically include mixing matrices referenced by A_k , bias vectors referenced by b_k , and pdf parameters referenced by β_k .

The A_k 's are N by M scalar matrices (called basis or mixing matrices) for the class k . N is the number of mixtures (e.g. sensors) and M is the number of sources, and it is assumed that $N \geq M$, as discussed above. The b_k 's are N -element bias vectors. The β_k 's are N -element pdf vectors. There are a total of K mixing matrices (A_1, \dots, A_K), K bias vectors (b_1, \dots, b_K), and K pdf vectors (β_1, \dots, β_K) that are learned as described herein.

Overview of the Unsupervised Adaptation and Classification Algorithm

Reference is now made to Fig. 2, which is a top-level flow chart that illustrates the unsupervised classification algorithm described herein. Due to the amount of information to be disclosed herein, many of the steps in the algorithm are referenced in Fig. 2 and then shown in detail in other Figures and discussed in detail with reference thereto. The unsupervised classification algorithm begins at a box 200 that indicates the beginning of the unsupervised classification algorithm.

In an initialization step shown at 210, parameters Θ are initialized to appropriate values. Particularly, the mixing matrices A_k , bias vectors b_k and the pdf vectors β_k are initialized for each class from 1 to K . K is the total number of classes, and K is typically greater than one. The class probability for each class is typically initialized to $1/K$, unless another probability is suggested.

In one example, the mixing matrices A_k are set to the identity matrix, which is a matrix whose diagonal elements are one and all other elements are zero. Small random values (e.g. noise) may be added to any of the elements, which advantageously makes the mixing matrices different for each class. The bias vectors b_k may be set to the mean of all data vectors x_i in the data set. Some small random values (e. g. noise) may be added to each of the elements of the bias vectors, which makes the bias vectors different for each class.

Some embodiments implement a generalized Gaussian model for the sources, described in detail elsewhere herein, which allows adaptation and classification using a plurality of pdfs that deviate from the standard Gaussian distribution. In one example, these pdfs continuously vary from a uniform distribution to a delta function. In such one embodiment, $\beta = 0$ corresponds to the standard Gaussian distribution, $\beta = -1$ corresponds to a uniform distribution, $\beta = 1$ corresponds to a Laplacian distribution, and $\beta \rightarrow \infty$ corresponds to the delta function. In this embodiment, the pdf parameters are initialized to a value between -1 and ∞ , although in other embodiments a different variable transformation can be utilized in accordance with the purposes of the generalized Gaussian model. The pdf parameter may be a random variable that is continuously varying to define a wide range of pdfs or the pdf parameter may be allowed only at a plurality of discrete intervals, such as at intervals of 0.1. Also, the pdf parameter may be limited to a range of -1 to 100, for example. The pdf vectors β_k are initialized to provide initial values for the pdf parameters for source (which corresponds to each basis function), and accordingly, the pdf parameters are initialized for each pdf vector β_k from $t = 1$ to T . The β parameters for each class are stored in a pdf vector comprising elements β_1, \dots, β_N , which is an N -element vector of pdf parameters.

In embodiments in which an extended informax algorithm has been implemented instead of the generalize Gaussian model, switching parameter vectors q_k should be initialized for each data vector from $t=1$ to T to designate a sub- or super-Gaussian distribution. Each of the switching vectors q_1, \dots, q_T are N -element switching parameter vectors used to create a diagonal matrix in operations performed in a classification algorithm described herein. In one embodiment the switching parameters $q_n \in \{1, -1\}$ respectively designate either a sub- or super-Gaussian probability distribution function (pdf).

At 220 the data vectors x_i for the data set (from $t = 1$ to $t = T$) are provided to the algorithm. The data index is t , and the number T is the total number of data vectors in the data set. Referring briefly to Fig. 1, it can be seen that in one embodiment each data vector x_i has N elements that correspond to the number of mixtures (linear combinations), which also correspond to the number of sensors. In some embodiments it may be convenient to process blocks of data instead of a single vector; for example, it may be useful to combine the data vectors from x_1 to x_{100} , and treat the combined vectors, for purposes of performing adaptation and/or classification, as a single block (single "vector") of data.

At 230 the main adaptation loop is performed to adapt the class parameters Θ of all the

classes. This is an iterative operation performed for each data vector in the data set, and then repeated until convergence, as described in more detail below with reference to Figs. 3, 4, 5, and 6. Generally, for each data vector the adaptation process in the main adaptation loop includes performing probabilistic calculations for each class, then adapting the class
 5 parameters based upon those calculations, and repeating these operations for each data vector. Until the algorithm converges, the main adaptation loop is repeated until the algorithm converges. Operations within the main adaptation loop will be described in detail with reference to Figs. 3, 4, 5, and 6.

At 240, after the main adaptation loop 230 has completed one loop, the probability of
 10 each class can be adapted using a suitable learning rule. In some embodiments, this operation will be performed only after several iterations of the main loop when the learning rate slows, or at other suitable points in the process as determined by the application. One suitable learning rule, performed for each class from $k = 1$ to $k = K$, is

$$p(C_k) = \frac{1}{T} \sum_{t=1}^T p(C_k | x_t, \Theta)$$

15 This calculation gives the adapted class probability for each class for the next operation. The adapted class probability is then used in the next iteration of the main adaptation loop. In other embodiments, other suitable learning rules could be used to adapt the class probabilities for each class.

At 250, the number of classes K may be adapted using a split and merge algorithm.
 20 One such algorithm, described with reference to Fig. 7 begins by assuming a certain number of classes (K), and performing a number of iterations of the main adaptation loop to calculate a first set of parameters Θ_1 . If all of the learned classes are sufficiently different, the assumed number of classes may adequately represent the data. However if two of the classes are very similar they may be merged. If all are different, and is possible that there may be more classes,
 25 then the number of classes (K) can be incremented, the main adaptation loop reiterated to calculate a second set of parameters Θ_2 , and the first and second sets of parameters compared to determine which more accurately represents the data. The adapted K value for the number of classes is then used in the next iteration of the main adaptation loop.

Another way of adapting the number of classes is to use a split and merge EM algorithm
 30 such as disclosed by Ueda, et al. in "SMEM Algorithm for Mixture Models", published in the Proceedings of the Advances in Neural Information Processing Systems 11, (Kearns et al., editors) MIT Press, Cambridge MA (1999), which overcomes the local maximum problem in parameter estimation of finite mixture models. In the split and merge EM algorithm described by Ueda et al., simultaneous split and merge operations are performed using a criterion that
 35 efficiently selects the split and merge candidates that are used in the next iteration.

At 260, the results of the previous iteration are evaluated and compared with previous iterations to determine if the algorithm has converged. For example, the learning rate could be

observed as the rate of change in the average likelihood of all classes:

$$p(\mathbf{X}|\Theta) = \prod_{t=1}^T p(\mathbf{x}_t|\Theta) = \prod_{t=1}^T \sum_{k=1}^K p(\mathbf{x}_t|C_k, \theta_k) p(C_k)$$

The main adaptation loop 230 and (if implemented) the class number and probability adaptation steps 240 and 250 will be repeated until convergence. Generally, to determine
 5 convergence the algorithm tests the amount of adaptation (learning) done in the most recent iteration of the main loop. If substantial learning has occurred, the loop is repeated. Convergence can be determined when the learning rate is small and stable over a number of iterations sufficient to provide a desired level of confidence that it has converged. If, for example, the change in the average likelihood is very small over several iterations, it may be
 10 determined that the loop has converged.

Determining when an algorithm has converged is very application-specific. The initial values for the parameters can be important, and therefore they should be selected carefully on a case-by-case basis. Furthermore, as is well known, care should be taken to avoid improperly stopping on a local maximum instead of at convergence.

15 After the loop has converged, then each data vector is assigned to one of the classes. Particularly, for $t = 1$ to $t = T$, each data vector \mathbf{x}_t is assigned to a class. Typically each data vector \mathbf{x}_t is assigned to the class with the highest probability, which is the maximum value of $p(C_k|\mathbf{x}_t, \Theta)$ for that data vector. In some embodiments, *a priori* knowledge may be used to improve accuracy of the assignment process; for example, if it is known that one of the classes
 20 (e.g. a mixed conversation), is likely to extend over a number of samples (e.g., a period of time), a number of adjacent data vectors (e.g. 100 or 2000 adjacent data vectors) can be grouped together for purposes of more accurately assigning the class.

Finally, at 280, it is indicated that all class parameters are known, and each observed data vector is now classified. The source data is now separated into its various sources and
 25 available for use as desired.

Description of the Main Adaptation Loop 230

Reference is now made to the flow chart of Fig. 3 in conjunction with the flow charts of Figs. 4, 5, and 6 to describe the main adaptation loop 230 shown in Fig. 2 and described briefly with reference thereto.

30 Operation begins at 300, where the flow chart indicates that the main adaptation loop will adapt the class parameters Θ (for all classes) responsive to all data vectors and previously computed (or assumed) parameters.

At 310, the data index t is initialized to 1, and then operation proceeds to 320 which is the initial calculation loop, then to 330 which is the class probability calculation, then to 340
 35 which is the mixing matrix adaptation loop, then to 350 which is the bias vector adaptation loop, and then to 355 which is the pdf vector adaptation loop. At 360, the data index is tested to determine if all of the data vectors (there are T) have been processed. If not, the data index

t is incremented at 460 and the loops 320, 330, 340, 350, and 355 are repeated. Operation in the main adaptation loop continues until each of the data vectors has been processed, at which point the data index t is equal to T , and the main adaptation loop is complete as indicated at 380.

5 Reference is now made to Fig. 4 to describe the initial calculation loop 320 of Fig. 3. Fig. 4 is a flow chart that begins at 400, illustrating the series of operations in the initial calculation loop. Briefly, for each class the source vector is calculated, the probability of that source vector is calculated, and the likelihood of the data vector given the parameters for that class is calculated. Although the box 320 suggests a single loop, in some embodiments, this
10 step could be implemented in two or three separate loops each loop completing K iterations.

At 410, the class index k is initialized to 1. At 420, a first calculation calculates the source vector $s_{t,k}$ which will be used in subsequent calculations. The source vector is computed by performing the following operations:

$$s_{t,k} = A_k^{-1} \cdot (x_t - b_k)$$

15 At 430, a second calculation calculates the probability of the source vector using an appropriate model. In one embodiment, the generalized Gaussian model is used to model a wide variety of pdfs, from a uniform distribution to a delta function, as described elsewhere herein. In one embodiment, the pdf parameter is a continuously-varying random variable β that is utilized to designate a particular probability density function. Typically, the standard
20 deviation σ is assumed (normalized) to be one; however, in some embodiments the standard deviation could be adapted in the calculation. The mean μ of the Gaussian pdfs is provided by the bias vector. Advantageously, using the generalized Gaussian model, a wide variety of statistical distributions can be characterized, which can improve classification accuracy and provide much more accurate modeling, more accurate codes, and therefore more accurate
25 classification or any other result that may be desired. Generally, it provides better performance, because it can better model the underlying statistical distribution of the data. The probability is calculated using the following equation:

$$\log p(s_{t,k}) \propto - \sum_{n=1}^N \eta |s_{t,k,n} - \mu_{k,n}|^{g-1} g_{k,n}(\beta) c_{k,n}(\beta) \sigma_{k,n}^{-g}$$

30 where $\eta = \text{sign}(s_{t,k,n} - \mu_{k,n})$. The functions $c(\beta)$ and $g(\beta)$ are defined elsewhere herein. Once the log is calculated, then the \log^{-1} is calculated using the exp function to give the desired probability.

An alternative embodiment of the algorithm utilizes an extended infomax model that allows only two pdfs. In this model, super-Gaussian densities are approximated by a density model with a "heavier" tail than the Gaussian density; and sub-Gaussian densities are
35 approximated by a bimodal density in accordance with an extended infomax algorithm as described by Lee et al., "Independent Component Analysis Using an Extended Infomax Algorithm for Mixed Subgaussian and Supergaussian Sources" Neural Computation 11, pp.

417-441 (1999). The log of the distribution is given by the following:

$$\log p(s_{i,k}) \propto - \sum_{n=1}^N \left(q_{n,k} \log [\cosh s_{i,k,n}] - \frac{s_{i,k,n}^2}{2} \right)$$

Once the log is calculated, then the \log^{-1} is calculated using the exp function to give the desired probability.

- 5 The switching parameter q_n , which is selected from the set of 1 and -1, is determined by whether the distribution is sub-Gaussian or super-Gaussian. For super-Gaussian distributions, the switching parameter is $q_n = 1$, and for sub-Gaussian distributions the switching parameter is $q_n = -1$.

- 10 As an alternative that is suitable for sparse representations (representations in which many of the source vectors are clustered around zero, the source probability can be computed using a simpler form:

$$\log p(s_{i,k}) \propto - \sum_{n=1}^N |s_{i,k,n}|$$

It may be noted that this simpler form does not require knowledge of the switching parameters q .

- 15 At 440, a third calculation calculates the likelihood of the data vector x_i given the parameters for class k :

$$p(x_i | \theta_k, C_k) = \frac{p(s_{i,k})}{\det[A_k]}$$

This likelihood is used in subsequent calculations.

- 20 At 450, the class index is tested to determine if the operations in the loop have been completed for each of the classes. If additional classes remain to be processed, the class index is incremented as indicated at 460, and the first, second and third operations 420, 430, and 440 are repeated for each subsequent class. After all classes have been processed, the initial calculation loop is complete as indicated at 470.

- 25 Referring again to Fig. 3, the class probability loop 330 is performed by calculating, from $k = 1$ to $k = K$, the following:

$$p(C_k | x_i, \Theta) = \frac{p(x_i | \theta_k, C_k) \cdot p(C_k)}{\sum_{k=1}^K p(x_i | \theta_k, C_k) \cdot p(C_k)}$$

The class probability loop requires all the data from the initial calculation loop 320 to calculate the sum in the denominator, and therefore cannot be calculated until after completion of the initial calculation loop.

- 30 Reference is now made to Fig. 5, which is a detailed flow chart of step 340 in Fig. 3, illustrating operations to adapt the mixing matrices for each class. The flow chart of Fig. 5 begins at 500, and at 510 the class index k is initialized to 1. An appropriate adaptation

algorithm is used, such as the gradient ascent-based algorithm disclosed by Bell et al. U.S. Patent 5,706, 402, which is incorporated by reference herein. The particular adaptation described herein includes an extension of Bell's algorithm in which the natural gradient is used as disclosed by Amari et al., "A New Learning Algorithm for Blind Signal Separation",
 5 *Advances in Neural Information Processing Systems 8*, pp. 757-763 (1996) and also disclosed by S. Amari, "Natural Gradient Works Efficiently in Learning", *Neural Computation*, Vol. 10 No. 2, pp. 251-276 (1998). Particularly, the natural gradient is also used in the extended infomax ICA algorithm discussed above with reference to step 430, which is able to blindly separate mixed sources with sub- and super-Gaussian distributions. However, in other
 10 embodiments other rules for adapting the mixing matrices could be used.

At 520, the gradient ΔA_k is used to adapt the mixing matrix for class k :

$$\Delta A_k \propto \frac{\partial}{\partial A_k} \log p(\mathbf{x}_i | \Theta) = p(C_k | \mathbf{x}_i, \Theta) \frac{\partial}{\partial A_k} \log p(\mathbf{x}_i | C_k, \theta_k)$$

For the generalized Gaussian mixture model, the preceding gradient can be approximated using an ICA algorithm like the following, and also includes the natural gradient:

$$15 \quad \Delta A_k \propto p(C_k | \mathbf{x}_i, \Theta) A_k [I - \phi(s_k) s_k^T]$$

where the score function is:

$$\phi(s_k) = - \frac{\partial}{\partial A_k} \log p(s_k)$$

The log function above has been previously determined. The gradient in the general update for A_k can be approximated using an ICA algorithm like the following extended infomax ICA
 20 learning rule, which applies generally to sub- and super-Gaussian source distributions, and also includes the natural gradient:

$$\Delta A_k \propto p(C_k | \mathbf{x}_i, \Theta) A_k [I - Q_k \tanh(s_k) s_k^T - s_k s_k^T]$$

where Q_k is an N -dimensional diagonal matrix whose switching parameters are q_n , specifically, $q_n = 1$ for super-Gaussian pdfs and $q_n = -1$ for sub-Gaussian pdfs.

25 In alternative embodiments, the gradient can also be summed over multiple data points, which is a technique that can be used to optimize the convergence speed.

When only sparse representations are needed, a Laplacian prior ($p(s) \propto \exp(-|s|)$) can be used to adapt the matrix, which advantageously eliminates the need for the switching parameters, and also simplifies the infomax learning rule described herein. The update for A_k
 30 is:

$$\Delta A_k \propto p(C_k | \mathbf{x}_i, \Theta) A_k [I - \text{sign}(s_k) s_k^T]$$

As an additional advantage, this simplified learning rule simplifies calculations for $p(s_k)$, as described above.

At 530, the new mixing matrix A_k is calculated by weighting the update ΔA_k by the
 35 class probability:

$$\mathbf{A}_k = \mathbf{A}_k^* + r \cdot p(C_k | \mathbf{x}_t, \Theta) \cdot \Delta \mathbf{A}_k$$

where \mathbf{A}_k^* is the previously-estimated value for the mixing matrix, and r is a predetermined adaptation rate smaller than 1.0, such as 0.2, chosen such that the solution converges at a reasonable rate.

5 At 540, the class index is tested to determine if the mixing matrices for each of the classes have been adapted. If one or more additional classes remain to be adapted, the class index is incremented as indicated at 550, and the adaptation operations 520, 525, and 530 are repeated for each additional class. After all classes have been adapted, the mixing matrix adaptation loop is complete, as indicated at 560.

10 Reference is now made to Fig. 6A, which is a detailed flow chart of step 350 (Fig. 3) that illustrates operations to adapt the bias vectors for each class k . The adaptation described below is based upon a maximum likelihood estimation learning rule to obtain the next value of the bias vector. However, in other embodiments other rules for adapting the bias vectors could be used.

15 The flow chart of Fig. 6A begins at 600, and at 610 the class index k is initialized to 1. At 620, the next value for the bias vector is calculated. An approximate EM update rule is:

$$\mathbf{b}_k = \frac{\sum_{t=1}^T \mathbf{x}_t p(C_k | \mathbf{x}_t, \Theta)}{\sum_{t=1}^T p(C_k | \mathbf{x}_t, \Theta)}$$

This rule provides the value of the bias vector \mathbf{b}_k that will be used in the next iteration of the main adaptation loop.

20 At 630, the class index is tested to determine if the bias vectors for each of the classes have been adapted. If one or more additional bias vectors remain to be adapted, the class index is incremented as indicated at 640, and the adaptation operations 620 are repeated for each additional class. After all classes have been adapted, the bias vector adaptation loop is complete, as indicated at 650.

25 Reference is now made to Fig. 6B, which is a detailed flow chart of step 355 (Fig. 3) that illustrates operations to adapt the pdf vectors for each class k . Using the generalized Gaussian model, a generalized Gaussian pdf is adapted for each of the sources (basis functions) in the class k using a suitable adaptation technique together with the learning rule discussed below. The learning rules for the adaptation of the β parameters for all sources assume a zero mean and a unit variance (i.e. $\mu = 0$ and $\sigma = 1$), leaving only the pdf parameter β to be adapted for each basis function. In other words, by adapting the elements of the pdf vector β_k , a generalized Gaussian pdf is specified for each source (and each basis function) for the class k . One adaptation described below is based upon a maximum posterior value learning rule to obtain the next value of the pdf parameters. However, in other embodiments other rules for

30 adapting the pdf parameters could be used.

35

The flow chart of Fig. 6B begins at 660, and at 665 the class index k is initialized to 1. At 670, this adaptation is performed by determining the partial derivative in the equation below using gradient ascent or a maximum posterior value estimation for the β vector of that class, which contains a β parameter for each source (basis function) in the class. In the
 5 adaptation of β the gradient of the component density with respect to β_k is weighted by the previously-computed class probability $p(C_k | \mathbf{x}_n, \Theta)$ to give the update to the pdf vector:

$$\Delta \beta_k \propto p(C_k | \mathbf{x}_n, \Theta) \frac{\partial}{\partial \beta_k} \log p(\mathbf{x}_n | C_k, \theta_k)$$

The class probability is known because it has been calculated earlier in the algorithm, as shown at 330 (Fig. 3). Therefore only the partial derivative with respect to β_k of the log of
 10 the probability of the data given the class parameters remains to be determined. This can be determined analytically using

$$\frac{\partial}{\partial \beta_k} \log p(\mathbf{x}_n | C_k, \theta_k) = \frac{\partial}{\partial \beta_k} \log p(s_{i,k})$$

The log of the source probability $p(s_{i,k})$ has been determined earlier in the algorithm at 430 (Fig. 4), and this information could be utilized to determine the gradient of the component density
 15 with respect to β_k .

An alternative way of estimating the gradient of the component density with respect to β_k is to use the maximum posterior value of β to estimate the partial derivative. Particularly, the posterior distribution of β , given the currently-calculated source vector $\mathbf{s} = \{s_1, \dots, s_N\}$ is

$$p(\beta_{k,n} | s_i) \propto p(s_i | \beta_{k,n}) \cdot p(\beta_{k,n})$$

20 where the data likelihood $p(s|\beta)$ is defined by

$$p(\mathbf{s}|\beta) = \prod_n \omega(\beta) \exp \left[-c(\beta) |s_n|^{2/(1+\beta)} \right]$$

The functions $\omega(\beta)$ and $c(\beta)$ are defined elsewhere herein. In this calculation, the value for $p(\beta_{k,n})$ is defined by the prior calculated distribution for the pdf parameter β . Because $\beta > -1$ (as defined herein), it is convenient to use $p(\beta_{k,n}) \sim \text{Gamma}(1 + \beta | a, b)$. The Gamma distribution
 25 is a well-known pdf that has a non-symmetrical form and requires two parameters: a and b . In one embodiment, calculation time of the adaptation process is reduced by choosing the values $a = 2$ and $b = 2$, which gives a broad prior distribution with a 95% density range of $[-0.5, 10.5]$, which is adequate for many uses.

At 680, for each pdf parameter in the pdf vector, the partial derivative calculated in the
 30 previous step is weighted by the class probability and an adaptation rate parameter, and then added to the previously computed value for the pdf parameter to compute the next value for the pdf parameter.

At 685, the class index is tested to determine if the pdf vectors for each of the classes have been adapted. If one or more additional pdf vectors remain to be adapted, the class
 35 index is incremented as indicated at 690, and the adaptation operations are repeated for each

additional class. After all classes have been adapted, the pdf vector adaptation loop is complete, as indicated at 695.

In an alternative embodiment that implements the extended infomax ICA model instead of the generalized Gaussian model, then the pdf parameter flow chart of Fig. 6B would be replaced by a system that adapts the switching parameter vector \mathbf{q}_i in any suitable manner. For example, the following learning rule can be used to update the N elements of the switching parameter vector, from $n = 1$ to $n = N$:

$$q_{k,n} = \text{sign}\left(E\{\text{sech}^2(s_{k,n})\}E\{s_{k,n}^2\} - E\{[\tanh(s_{k,n})]s_{k,n}\}\right)$$

Reference is now made to Fig. 7 which is a flow chart of one method to adapt the number of classes as referenced by step 250 (Fig. 2); however, other methods of class adaptation are possible. Operation starts at block 700. At step 710, K (the number of classes) is initialized to an appropriate value. In one embodiment, K may be initially set to one, in other embodiments K may be guessed to a conservative estimate. Next, at step 720 the main adaptation loop is performed at least a predetermined number of iterations to obtain parameters θ_k for each class. In step 720, it may be sufficient to stop before convergence, depending upon the data and the application.

After the parameters are obtained, at step 730 the class parameters are compared. At branch 740, if two or more classes are similar, operation branches to step 750 where similar classes are merged, and the class adaptation operation is complete as indicated at 760. However, returning to the branch 740, if two or more classes are not similar, then K is incremented and the parameters are initialized for the new class. The new parameters may be initialized to values similar to one of the other classes, but with small random values added. The operations 720 and 730 are repeated to adapt the class parameters for the new class number K , starting with the newly initialized class parameters and the previously learned class parameters. Then, at branch 740 the appropriate branch is taken, which either completes the operation or again increments the number of classes and repeats the loop.

Experimental Results and Implementations

Reference is now made to Fig. 8. In one experiment random data was generated from four different classes, and the algorithm described herein was used to learn the parameters and classify the data. The data points for the two classes in two-dimensional space were initially unlabeled. Each class was generated using random choices for the class parameters. Each data point represented a data vector of the form $\mathbf{x}_i = (x_1, x_2)$. The goal for the algorithm was to learn the four mixing matrices and bias vectors given only the unlabeled two-dimensional data set.

In the experiment, the parameters were randomly initialized, and the algorithm described in Fig. 2, including the main adaptation loop was performed. The algorithm converged after about 300 iterations of the main adaptation loop, and in Fig. 8, the arrows 801,

802, 803, and 804 are indicative of the respective mixing matrices A_1 , A_2 , A_3 , and A_4 , and bias vectors b_1 , b_2 , b_3 , and b_4 . The arrows show that the parameters were learned correctly. In this experiment, the classes had several overlapping areas, and the classification error on the whole data set was calculated at about 5.5%. In comparison, the Gaussian mixture model used in the Autoclass algorithm gave an error of about 7.5%. The Autoclass algorithm is disclosed by Stutz and Cheeseman, "Autoclass—a Bayesian Approach to Classification" Maximum Entropy and Bayesian Methods, Kluwer Academic Publishers (1994). For the k-means clustering algorithm (i.e., Euclidean distance measure) the error was calculated at about 11.3%.

Reference is now made to Figs. 9A, 9B, 9C, 9D, 9E, 9F, and 9G, which represents raw and processed data for a experiment in which two microphones were placed in a room to record a conversation between two persons with music in the background. The conversation between the two persons is in an alternating manner in which a first person talks while a second person listens (giving a first class), and then the second person talks while the other person listens (giving a second class). The time at which one speaker stops speaking and other begins speaking is unknown. The goal is to determine who is speaking, separate the speaker's voice from the background music and recover their conversation.

In Fig. 9A, the first microphone provides a first channel of raw mixed data designated x_1 , and in Fig. 9B the second channel provides a second channel of raw data designated by x_2 . Each channel receives the alternating voices of the first and second persons together with the background music. The horizontal axis shows time intervals (in seconds). In one experiment, the data included 11 seconds of data sampled at a rate of 8 kHz. The vertical axis shows amplitude about a reference value.

In this example there two classes ($K=2$). The adaptation algorithm described with reference to Fig. 2 was used to adapt two mixing matrices and two bias vectors to the two classes. A first mixing matrix A_1 and a first bias vector b_1 were randomly initialized and adapted to define the first class in which the first person's voice is combined with the background music, and a second mixing matrix A_2 and a second bias vector b_2 were randomly initialized and adapted to define the second class in which the second person's voice is combined with the background music. For each matrix adaptation step, a step size was computed as a function of the amplitude of the basis vectors in the mixing matrix and the number of iterations.

Figs. 9C and 9D show the source signals after adaptation, classification, and separation using a block size of 2000 samples to improve accuracy of the classification, as discussed below. Fig. 9C shows the time course of the two speech signals with markers that correctly indicate which speaker is talking. Particularly, the first speaker is speaking at time intervals 910, 912, and 914, and the second speaker is speaking at time intervals 920, 922, 924. Fig. 9D shows the time course of the background music.

In this example, a single sample typically did not include enough information to

unambiguously assign class membership. Fig. 9E shows the class conditional probability $p(C_2|x_i, \theta_2) = 1 - p(C_1|x_i, \theta_1)$. Fig. 9E shows many values clustered around 0.5, which indicates uncertainty about the class membership of the corresponding data vectors using a single sample. Using a threshold of 0.5 to determine the class membership for single samples as shown in Fig. 9E gives an error of about 27.4%. In order to improve accuracy of assignment to classes, the prior knowledge that a given class will likely persist over many samples was used. In some embodiments this a priori knowledge is incorporated into a complex temporal model for $p(C_k)$; however, in this experiment the simple procedure of computing the class membership probability for an n-sample block was used. Fig. 9F shows the results for a block size of 100 samples, which provided an error rate of only about 6.5%, thereby providing a much more accurate estimate of class membership. When a sample block size of 2000 was used, as shown in Fig. 9G, the error rate dropped to about 0.0%, and the class probabilities were recovered and matched those in Fig. 9C.

For this experiment, the SNR (Signal to Noise Ratio) with a block size of 100 samples was calculated to be 20.8 dB and 21.8 for classes 1 and 2, respectively. In comparison, a standard ICA algorithm using infomax, which was able to learn only one class, provided a SNR of only 8.3 dB and 6.5 dB, respectively.

Implementations

Reference is now made to Fig. 10. Generally, the adaptation and classification algorithms described herein, such as the algorithm shown in Fig. 2, will be implemented in a computational device such as a general purpose computer 1010 that is suitable for the computational needs of the algorithm. In some embodiments it may be implemented in an ASIC (application specific integrated circuit) for reasons such as low-cost and/or higher processing speed. Due to the computational requirements of the algorithm, it may be advantageous to utilize a computer with a fast processor, lots of memory, and appropriate software.

The adaptation and classification algorithms described herein can be used in a wide variety of data processing applications, such as processing speech, sound, text, images, video, text, medical recordings, antenna receptions, and others. For purposes of illustration of the variety of data that can be adapted and classified by this algorithm, Fig. 10 shows that speech, sound, images, text, medical data, antenna data, and other source data may be input into the computer 1010. The text may be in computer-readable format, or it may be embedded in an image. The data may be generated by, or stored in another computer shown at 1015. Depending upon the sensor(s) used, the raw data may already have the form of digital data. If not, a digital sampler 1020 can be used to digitize analog data or otherwise to process it as necessary to form suitable digital data. The output from the computer can be used for any suitable purpose or displayed by any suitable system such as a monitor 1025 or a printer 1030.

The data set can be processed in a variety of ways that specifically depend upon the

data set and the intended application. For purposes of description, data processing generally falls into two categories: 1) a first category in which unknown parameters for multiple classes are adapted from the data to find unknown structure in data, for example for separation of sources, and 2) a second category in which the unknown class parameters for multiple classes are adapted using a training set, and then the adapted class parameters for each class are used (and sometimes re-used) to find the certain or selected structure in the data. However, because the categories are chosen only for descriptive purposes, some uses may fall into both categories.

The first category, in which a mixing matrix is adapted from the data and then the sources are separated, is illustrated in the flow chart of Fig. 2 and is described with reference thereto. An example of this first category is speech enhancement, such as the microphone mixing example disclosed with reference to Figs 9A-9G, in which parameters for two classes are adapted for the purpose of classifying mixed data to separate two voices from the background music.

Another example of the first category is medical data processing. EEG (Electroencephalography) recordings are generated by multiple sensors each of which provides mixed signals indicative of brain wave activity. A person's brain wave activity transitions through a number of different cycles, such as different sleep levels. In one embodiment the adaptation and classification algorithm of Fig. 2 could be used to adapt the class parameters for multiple classes, to classify the data, and to separate the sources. Such an implementation could be useful to monitor normal activity as well as to reject unwanted artifacts. An additional medical processing application is MRI (Magnetic Resonance Imaging), from which data can be adapted and classified as described in Fig. 2.

Still another example of the first category is antenna reception from an array of antennas, each operating as a sensor. The data from the each element of the array provides mixed signals that could be adapted, classified, and separated as in Fig. 2.

Figs. 11 and 12 illustrate the second category of data processing in which the system is trained to learn mixing matrices, which are then used to classify data. Fig. 11 is a flow chart that shows the training algorithm beginning at 1100. At step 1110 the training data is selected; for example image data such as nature scenes and text can be selected to provide two different classes. At step 1120 the parameters are initialized and the training data vectors are input in a manner such as described with reference to steps 210 and 220 of Fig. 2. Steps 1130, 1140, 1150, and 1160 form a loop that corresponds to the steps 230, 240, 250, and 260 in Fig. 2, which are described in detail with reference thereto. Briefly, step 1130 is the main adaptation loop shown in Fig. 3 wherein the mixing matrices and bias vectors are adapted in one loop through the data set. Step 1140 is the step wherein the probability of each class is adapted from 1 to K. Step 1150 is the optional step wherein the number of classes may be adapted. At step 1160 the results of the previous iteration are evaluated and compared with previous

iterations to determine if the algorithm has converged as described in more detail with reference to step 260 of to Fig. 2. After convergence, operation moves to block 1170 wherein the final classes A_k and bias vectors b_k for each class from 1 to K are available.

Fig. 12 is a flow chart that shows the classification algorithm beginning at 1200. At step 1210 the data vectors in the data set are collected or retrieved from memory. At step 1220 the data index t is initialized to 1 to begin the loop that includes the steps 1230, 1240, 1250, and the decision 1260. At step 1225 the adapted class parameters (from step 1170) previously computed in Fig. 11 are inserted into the loop via step 1230. The step 1230 is the initial calculation loop shown in Fig. 4 and described with reference thereto, wherein using the previously-adapted class parameters, the source vector is calculated, the probability of the source vector is calculated, and the likelihood of the data vector given the parameters for that class is calculated. The step 1240 is step 330 of Fig. 4, wherein the class probability for each class is calculated. At step 1250 each data vector is assigned to one of the classes. Typically the class with the highest probability for that data vector is assigned or a priori knowledge can be used to group the data vectors and thereby provide greater accuracy of classification. As shown at 1260 and 1270, the loop is repeated for all the data vectors, until at 1280 classification is complete and additionally the source vectors, which have been computed in previous steps, are available if needed. The classified data can now be used as appropriate. In some instances, the classification information will be sufficient, in other instances the source vectors together with the classification will be useful.

In some embodiments, all the basis functions (i.e. the column vectors of the mixing matrix) will be used to classify the data in Fig. 12. In other embodiments, less than all of the basis vectors may be used. For example, if $N=100$, then the 30 basis vectors having the largest contribution could be selected to be used in calculations to computed the class probability.

One advantage of separating the adaptation algorithm from the classification process is to reduce the computational burden of the algorithm. The adaptation algorithm requires a huge number of computations in its many iterations to adapt the mixing matrices and bias vectors to the data. Furthermore, in some instances expert assistance may be required to properly adapt the data. However, once the class parameters have been learned, the classification algorithm is a straightforward calculation that consumes much less computational power (i.e. less time). Therefore, implementing a classification algorithm as in Fig. 12 using previously learned class parameters as in Fig. 11 is typically more practical and much less costly than implementing a complete adaptation and classification system such as shown in Fig. 2.

Fig. 13 is a diagram that illustrates encoding an image 1300 (shown in block form). The image is defined by a plurality of pixels arranged in rows and columns (e.g. 640x480), each pixel having digital data associated therewith such as intensity and/or color. The pixel data is supplied by a digital camera or any other suitable source of digital image data. A plurality of

patches 1310 are selected from the image, each patch having a predefined pixel area, such as 8x8, 12x12, or 8x12. To illustrate how the data vectors are constructed from the image data, an expanded view of patch 1310a shows a 3x3 pixel grid. Each of the nine pixels within the 3x3 supplies one of the 9-elements of the data vector x_i in a pre-defined order. Each of the patches likewise forms a data vector. Referring now to Fig. 11, the data vectors are used as training data at 1110 to adapt the mixing matrices and bias vectors to provide the class parameters, including the trained mixing matrices and bias vectors, as illustrated at 1170. The image is encoded by the adapted class parameters.

The selection process to determine which patches 1310 will be selected depends upon the embodiment. Generally, a sufficient number and type of patches should be selected with a sufficient pixel count to allow adequate adaptation of mixing matrices and bias vectors for each class of interest. In some cases the patches will be randomly selected, in other cases the patches will be selected based upon some criteria such as their content or location.

Image classification is, broadly speaking, the process of encoding an image and classifying features in the image. The class parameters may be learned from a particular image, as in segmentation described below, or they may be learned from a training set that is adapted from certain selected classes of interest. For example text and nature images may be encoded to provide parameters for the two classes of nature and text. Using the learned parameters, the classification algorithm (Fig. 12) is then performed to classify the image data.

In order to collect the data for the classification process (step 1210 of Fig. 12) a blockwise classification may be performed in which the image is divided into a grid of contiguous blocks, each having a size equal to the patch size. Alternatively, in a pixelwise classification a series of overlapping blocks are classified, each block being separated by one pixel. The pixelwise classification will typically be more accurate than the blockwise classification, at the expense of additional computational time.

Segmentation is a process in which an image is processed for the purpose of finding structure (e.g. objects) that may not be readily apparent. To perform segmentation of an image, a large number of patches are selected randomly from the image and then used as training data at step 1110 (Fig. 11) in the adaptation (training) algorithm of Fig. 11, in order to learn multiple class parameters and thereby encode the image. Using the learned parameters, the classification algorithm (Fig. 12) is then performed to classify the image data. The classified image data can be utilized to locate areas that have similar structure. The classified data vectors may be further processed as appropriate or desired.

Other image classification processes may be employed for image recognition, in which an image is processed to search for certain previously learned classes. Reference is now made to Fig. 14, which is a view of an image that has been selectively divided into four distinct regions 1401, 1402, 1403, and 1404, each region having features different from the other three regions. Four different images could also be used, each image providing one of the regions.

For example four different types of fabric may be sampled, each region being a single type of fabric distinct from the others. A number of random samples are taken from each of the four regions, sufficient to characterize the distinct features within each region. In some embodiments the samples may be taken randomly from the entire image including the four regions, or from each of the four regions separately. However, if the regions are known, then it may be advantageous to sample patches from selected areas. In one example, a first group of samples 1411 are taken from the first region, a second group of samples 1412 are taken from the second region, a third group of samples 1413 are taken from the third region, and a fourth group of samples 1414 are taken from the fourth region. The samples are then used in the adaptation algorithm of Fig. 11 to adapt (learn) parameters for four classes, each of the four classes corresponding to the features of the four regions. If the classification is known in advance, the four classes may be adapted separately in four single-class adaptation processes.

The adapted parameters can then be used in the classification algorithm of Fig. 12 to classify regions within images that comprise an unknown combination of the four regions. One use is for locating and classifying bar codes that are placed arbitrarily upon a box. Four class parameters can be adapted (learned) in Fig. 11, including three classes corresponding to three different types of bar codes and a fourth class corresponding the typical features of the surrounding areas (noise). The adapted parameters for the four classes are then used in the classification algorithm of Fig. 12. The classified data and its corresponding data index provides the location and type of each bar code. Using this information, the bar code can then be read with a bar code reader suitable for that class, and the information in the bar code can be used as appropriate.

Image compression can be described using the steps described in Figs 11 and 12. The adaptation algorithm of Fig. 11 is first utilized to learn class parameters. In some embodiments, the class parameters are optimized, but in other embodiments, the class parameters may be learned using the particular image to be compressed. For standardized image systems, it is useful to optimize class parameters and provide the optimized parameters to both the person compressing of the image and the receiver of the compressed image. Such systems can have wide application; for example the JPEG compression system in wide use on the Internet utilizes an optimized algorithm that is known to the sender and the receiver of the compressed image.

Referring to Fig. 12, the image to be compressed is classified using the appropriate class parameters. The source vectors, which have been computed in Fig. 12, typically are clustered around zero, as shown in Fig. 15 at 1500. Because the source vectors that are near zero contain little information, they may be discarded. In other words, the source vectors between an upper value 1510 and a lower value 1520 may be discarded. The upper and lower values are selected dependent upon the implementation, taking into account such factors as how much information is desired to be transmitted and the bandwidth available to transmit the

image data. The compressed image data includes all source vectors above the upper value 1510 and below the lower value 1520 and the data index of the corresponding data vector in the image, together with information about the class to which each source vector belongs and the class parameters.

- 5 Other image processing applications include image enhancement, which includes de-noising and processes for reconstructing images with missing data. To enhance an image, the calculated source vectors are transformed into a distribution that has an expected shape. One such algorithm is disclosed by Lewicki and Sejnowski, "Learning Nonlinear Overcomplete Representations for Efficient Coding", Proceedings of Advances in Neural Information Processing Systems 10, (1998) MIT Press, Cambridge MA, pp. 556-562. Briefly, each image patch is assumed to be a linear combination of the basis functions plus additive noise: $\mathbf{x}_i = \mathbf{A}_k \mathbf{s}_k + \mathbf{n}$. The goal is to infer the class probability of the image patch as well as to infer the source vectors for each class that generate the image. The source vector \mathbf{s}_k can be inferred by maximizing the conditional probability density for each class:

$$15 \quad \hat{\mathbf{s}}_k = \min_i \left[\frac{\lambda_k}{2} \|\mathbf{x}_i - \mathbf{A}_k \mathbf{s}_k\|^2 + \alpha_k^T \|\mathbf{s}_k\| \right]$$

where α_k is the width of the Laplacian pdf and $\lambda_k = 1/\sigma_{k,n}^2$ is the precision of the noise for each class. The image is then reconstructed using the newly computed source vectors.

- A combination of image processing methods may be used for some implementations. For example satellite data processing may use image classification to look for certain structures such as mountains or weather patterns. Other embodiments may use segmentation to look for structure not readily apparent. Also, the satellite data processing system may use image enhancement techniques to reduce noise in the image.

- Speech processing is an area in which the mixture algorithms described herein have many applications. Speech enhancement, which is one speech processing application, has been described above with reference to Fig. 8. Other applications include speech recognition, speaker identification, speech/sound classification, and speech compression.

- Fig. 16 shows one system for digitizing and organizing speech data into a plurality of data vectors. A speaker 1600 generates sound waves 1610 that are received by a microphone 1620. The output from the microphone is digital data 1630 that is sampled a predetermined sampling rate such as 8 kHz. The digital data 1630 includes a series of samples over time, which are organized into data vectors. For example 100 sequential samples may provide the data elements for one data vector \mathbf{x}_i . Other embodiments may use longer data vectors, for example 500 or 1000 sample elements. In some embodiments the data vectors are defined in a series of contiguous blocks, one after the other. In other embodiments the data vectors may be defined in an overlapping manner; for example a first data vector includes samples 1 to 500, a second data vector includes samples 250 to 750, and so forth.

A speech recognition system first utilizes the adaptation (training) algorithm of Fig. 11

to adapt class parameters to selected words (or phonics), for the purpose of each word (or phonic) being a different class. For example, the adaptation algorithm may be trained with a word (or phonic) spoken in a number of different ways. The resulting class parameters are then used in the classification algorithm of Fig. 12 to classify speech data from an arbitrary speaker.

5 Once the speech has been classified, the corresponding class provides the word that is recognized by the system. The word can then be saved as text in a computer, for example.

Speech and sound classification systems utilize the adaptation (training) algorithm of Fig. 11 to adapt class parameters to selected features of speech or sound. For example, a language classification system adapts class parameters using the adaptation algorithm of Fig. 12
10 to distinguish between languages, for example, in such a manner that one language is represented by a first class and a second language is represented by another. The adapted class parameters are then used in the classification algorithm of Fig. 12 to classify speech data by language.

A speaker identification system adapts the class parameters to distinguish between the
15 speech of one person and the speech of another. The adapted class parameters are then used in the classification algorithm to identify speech data and associate it with the speaker.

A musical feature classification system adapts the class parameters to recognize a musical feature, for example to distinguish between musical instruments or combinations of musical instruments. The adapted class parameters are then used to classify musical data.

20 Speech compression is similar to image compression described above. A speech compression system uses adapted class parameters to classify speech data. Typically a speech compression system would use class parameters that are highly optimized for the particular type speech; however some embodiments may adapt the class parameters to the particular speech data. The speech data is classified as in Fig. 11 using the adapted class parameters.
25 The source vectors corresponding to the speech data, which have been computed during classification, are typically clustered around zero as shown in Fig. 15. Because the source vectors that are near zero contain little information, they may be discarded. In other words, the source vectors between an upper value 1510 and a lower value 1520 may be discarded. The upper and lower values are selected dependent upon the implementation, taking into
30 account such factors as how much information is desired to be transmitted and the available bandwidth. The compressed speech data includes all source vectors above the upper value 1510 and below the lower value 1520 and an identification of the time position of the corresponding data vector, together with information about the class to which each source vector belongs and the class parameters.

35 It will be appreciated by those skilled in the art, in view of these teachings, that alternative embodiments may be implemented without deviating from the spirit or scope of the invention. For example, the system could be implemented in an information retrieval system in which the class parameters have been adapted to search for certain types of information or

documents, such as books about nature, books about people and so forth. Also, in some embodiments some of the basis functions (less than all) can be selected from the adapted mixing matrix and used to classify data. This invention is to be limited only by the following claims, which include all such embodiments and modifications when viewed in conjunction with the above specification and accompanying drawings.

Description of the Generalized Gaussian Model

A generalized Gaussian model utilizes an exponential distribution that is a function of g to define distributions that deviate from the normal, standard Gaussian distribution. In its simplest form, this exponential distribution is

$$p(x) \propto \exp\left(-\frac{1}{2}|x|^g\right)$$

For convenience the variable g is often transformed into a function of β :

$$g = \frac{2}{1 + \beta}$$

By continuously varying β , it is possible to describe a plurality of pdfs including a normal Gaussian distribution ($\beta = 0$), sub-Gaussian pdfs ($\beta < 0$), and super-Gaussian pdf ($\beta > 0$).

The exponential distribution has been expressed by Box, G and Tiao, G., Bayesian Inference in Statistical Analysis, John Wiley and Sons (1973) in the following general form:

$$p(x|\mu, \sigma, \beta) = \frac{\omega(\beta)}{\sigma} \exp\left[-c(\beta)\left|\frac{x - \mu}{\sigma}\right|^{2/(1 + \beta)}\right]$$

where

$$c(\beta) = \left[\frac{\Gamma[\frac{3}{2}(1 + \beta)]}{\Gamma[\frac{1}{2}(1 + \beta)]} \right]^{1/(1 + \beta)}$$

where the Γ function is a well-known mathematical function, and

$$\omega(\beta) = \frac{\Gamma[\frac{3}{2}(1 + \beta)]^{1/2}}{(1 + \beta)\Gamma[\frac{1}{2}(1 + \beta)]^{3/2}}, \sigma > 0$$

In this form, the data's mean is given by μ and its standard deviation is given by σ . The pdf parameter β is a measure of kurtosis and helps define the distribution's deviation from normality. When $\beta = 0$, the distribution is the standard normal Gaussian distribution; at $\beta = 1$ it is a Laplacian (or double exponential). As $\beta \rightarrow -1$, the distribution becomes uniform over the unit interval. As $\beta \rightarrow \infty$, the distribution becomes a delta function at zero. The parameter β can also be converted to the standard kurtosis measure $\gamma_2 = E(x - \mu)^4 / \sigma^4 - 3$. For the exponential power distribution, this relation is

$$\gamma_2 = \frac{\Gamma[\frac{3}{2}(1 + \beta)]\Gamma[\frac{1}{2}(1 + \beta)]}{\Gamma[\frac{3}{2}(1 + \beta)]^2} - 3$$

Figs. 17A-17F show examples of the generalized Gaussian (exponential power) distribution for various values of β and the corresponding values of γ_2 . Fig. 17A shows a mostly

uniform distribution, while at the other spectrum Fig. 17F show a very narrow, pointed distribution. The following table shows the values of the pdf shown in Figs 17A-17F.

Table of values shown in Figs. 17A-17E

<u>Figure</u>	<u>Reference</u>	<u>β</u>	<u>γ</u>
Fig. 17A	1701	-0.75	-1.08
Fig. 17B	1702	-0.25	-0.45
Fig. 17C	1703	0.00	0.00
Fig. 17D	1704	+0.50	+1.21
Fig. 17E	1705	+1.00	+3
Fig. 17F	1706	+2.00	+9.26

The standard normal pdf is shown at 1703 in Fig. 17C, the Laplacian pdf is shown at 1705 in Fig. 17E. In addition, a function that approximates the ICA tanh function is shown at 1704 in Fig. 17D, which corresponds to the best-fitting exponential power distribution of the implied prior distribution under the widely-used tanh non-linearity in ICA.

Fig. 18 is a group of graphical depictions that show an example of fitting a two-dimensional distribution using generalized Gaussian source models (the ICA-exponential power model). In this example, the pdf parameters β were estimated periodically during learning by maximizing the posterior using the learning rule as described herein. Fig. 18A is a scatter plot that shows a two-dimensional data distribution shown generally at 1800 that includes a first distribution having $\beta = -1$ along a first axis and a $\beta = +4$ along a second axis. The arrows 1801 and 1802 indicate the learned basis functions, rescaled in length for plotting purposes. Fig. 18B and Fig. 18C are histograms of the distribution of the coefficients along each of the found axes; particularly, Fig. 18B shows a first histogram 1811 that illustrates the first distribution of the coefficients along the first axis 1801, and Fig. 18C shows a second histogram 1812 that illustrates the second distribution of coefficients along the second axis 1802. Figs. 18D and 18E show the learned values of the exponential power parameter β along each of the axes. Particularly, Fig. 18D shows an approximately uniform pdf 1821 corresponding to the first axis 1801. Fig. 18E shows a narrow, pointed pdf 1822 corresponding to the second axis 1802. The inferred (learned) pdf parameter β is -0.89 for the distribution along the first axis (Fig. 18D), while the inferred (learned) pdf parameter β is $+3.78$ for the distribution along the second axis (Fig. 18E), which is close the actual values of -1 and $+4$, respectively. This example shows a mixture of super- and sub-Gaussian sources, and shows how the distributions can be learned using a generalized Gaussian model as described herein.

Experimental Results Using Four Class Mixture

Fig. 19 is a graphical depiction of experimental results of classification and adaptation using a simulated mixture of four different classes, demonstrating the performance of the

generalized Gaussian mixture model described herein. Each of the classes was generated by two independent sources and bias vectors. The data in each class was generated by random choices for the parameters (β_k , A_k , and b_k) the β parameters were chosen as follows in the range from -1 to +2, resulting in the following densities, uniform, Gaussian, and heavy Laplacian.

- 5 The four classes included a first, a second, a third, and a fourth group shown generally at 1901, 1902, 1903, and 1904 respectively. The generalized Gaussian mixture model was used to learn the parameters and to classify the data. The goal was for the algorithm to learn the four basis vectors, the four bias vectors, and the pdf parameters β_k given the only unlabeled two-dimensional data set. In the initial algorithm, the parameters were randomly initialized. It was
10 found that the algorithm always converged after between about 300 to 500 iterations, depending primarily upon the initial conditions. During the adaptation process, the data log likelihood increased with the number of iterations. The arrow pairs 1911, 1912, 1913, and 1914 indicate the found basis vectors of A_k . Particularly, the first arrow set 1911 shows a first class, a second arrow pair 1912 shows a second class, a third arrow pair 1913 shows a third
15 class, and a fourth arrow pair 1914 shows a fourth class.

The following table shows the inferred (learned) parameters β_k and the Kullbak-Leibler divergence measure between the inferred density model and the actual source density:

β & (KL)	C_1	C_2	C_3	C_4
$\beta(s_k)$	-0.3	-0.2	1.6	2
KL ($p(s_k) q(\beta_k)$)	0.003	0.005	0.005	0.007

- Performance of the classification process was tested by processing each data instance with the learned parameters β_k , A_k , and b_k . The class probability was computed and the
20 corresponding instance label was compared to the highest class probability. In this example, in which the classes had several overlapping areas, the algorithm was repeated ten times with random initial conditions, and it converged in each instance. The difference between the computed β_k and the true β_k was generally less than 10%. Classification error on the whole data set averaged over ten trials was $4.0\% \pm 0.5\%$. In comparison, the Gaussian mixture
25 model described in AUTOCLASS (as disclosed by Stutze, J. and Cheeseman, P., "Autoclass - a Bayesian Approach to Classification" *Maximum Entropy and Bayesian Methods*, Kluwer Academic Publishers, 1994) gives an error of $5.5\% \pm 0.3\%$, and converged in all trials. Comparing these results with the k-means (Euclidean distance measure) clustering algorithm, the error was 18.3%, whereas the classification error with the original parameters was 3.3%.

CLAIMS

WHAT IS CLAIMED IS:

1. A computer-implemented method utilizing a generalized Gaussian mixture model to adapt class parameters and classify a plurality of data vectors having N elements using a plurality of generalized Gaussian probability density functions (pdfs) comprising:
 - receiving a plurality of data vectors \mathbf{x}_t from data index $t = 1$ to $t = T$;
 - initializing parameters for each class, including
 - K , the number of classes,
 - $p(C_k)$, the probability that a random data vector will be in class k ,
 - \mathbf{A}_k , a mixing matrix for each class k that defines a plurality of basis functions,
 - \mathbf{b}_k , a bias vector of bias parameters for said basis functions in the mixing matrix for the class k , and
 - $\boldsymbol{\beta}_k$, a vector of pdf parameters for said sources in the mixing matrix for the class k , each of said pdf parameters indicative of one of a plurality of generalized Gaussian pdfs; and
 - in a main adaptation loop, for each data vector \mathbf{x}_t from data index $t = 1$ to $t = T$, adapting the class parameters, comprising
 - adapting a mixing matrix \mathbf{A}_k for each class k , including adapting $\Delta \mathbf{A}_k$ by an ICA algorithm, and
 - adapting a bias vector \mathbf{b}_k for each class from class index $k = 1$ to $k = K$;
 - adapting a pdf vector $\boldsymbol{\beta}_k$ for each class from class index $k = 1$ to $k = K$;
 - repeating the main adaptation loop a plurality of iterations while observing a learning rate at each subsequent iteration; and
 - after observing convergence of said learning rate, then assigning each data vector to one of said classes.
2. The method of claim 1 wherein said step of adapting each of said pdf vectors includes determining the maximum posteriori value for each of the pdf parameters within said pdf vectors.
3. The method of claim 1 further comprising the step of utilizing the source vectors for the assigned classes to separate source signals in each of said classes.
4. The method of claim 1 further comprising the step of adapting the probability of each class after at least one iteration of said main adaptation loop.

5. The method of claim 1 further comprising the step of adapting the number of classes after at least one iteration of said main adaptation loop.
6. The method of claim 1 further comprising the step of supplying additional data vectors and utilizing the adapted class parameters to classify each of said additional data vectors into one of said k classes.
7. The method of claim 6 further comprising the step of utilizing the adapted class parameters to estimate source data densities.
8. The method of claim 6 further comprising the step of utilizing the adapted class parameters to encode source data.
9. The method of claim 8 further comprising the step of using said adapted class parameters to classify said encoded source data.
10. The method of claim 8 wherein said source data comprises image data indicative of an image, and further comprising the step of using said adapted class parameters to classify said encoded image data and reconstruct said image.
11. The method of claim 8 wherein said source data comprises speech data, and further comprising the step of using said adapted class parameters to classify said encoded speech data and reconstruct said speech pattern.
12. A computer-implemented method of classifying a plurality of data vectors using known class parameters including a predetermined mixing matrix for each class that includes a plurality of basis functions and a pdf parameter for each basis function that is indicative of one of a plurality of generalized Gaussian probability density functions, comprising:
 - receiving a plurality of data vectors x_t from data index $t = 1$ to $t = T$;
 - calculating the class probability for each data vector responsive to class parameters for each class including
 - K , the number of classes,
 - $p(C_k)$, the probability that a random data vector will be in class k ,
 - A_k , a mixing matrix for each class k that defines a plurality of basis functions,
 - b_k , a bias vector of bias parameters for each of said basis functions in the mixing matrix for the class k , and
 - β_k , a vector of pdf parameters for each of said basis functions in the

mixing matrix for the class k , each of said pdf parameters indicative of one of a plurality of generalized Gaussian pdfs; and
 assigning each data vector to one of said classes responsive to said calculated class probability.

- 5 13. The method of claim 12 wherein said step of calculating the class probability for each data vector comprises:
- performing an initial calculation loop for each class, including calculating a source vector $s_{k,i}$ responsive to said class parameters, a probability of the source vector $p(s_{k,i})$ and the likelihood of the data $p(x_i | \theta_k, C_k)$; and
- 10 performing a class probability loop that determines the class probability for each class $p(C_k | x_i, \Theta)$ for each data vector.
14. The method of claim 12 further comprising the steps of receiving data vectors from an image, classifying said data vectors using said predetermined mixing matrices, and then reconstructing said image.
- 15 15. The method of claim 12 further comprising the steps of receiving data vectors from a speech source, said data vectors being provided by a series of samples over time, and then classifying said data vectors using said predetermined mixing matrices.
16. A computer-implemented method that utilizes a generalized Gaussian mixture model to separate sources that each have an initially unknown pdf, said sources providing signals to a plurality of N sensors, each of the N sensors providing a mixed signal that is included in an N -element data vector, comprising:
- 20 receiving a plurality of data vectors x_t from data index $t = 1$ to $t = T$;
- initializing parameters including
- K , the number of classes,
- 25 $p(C_k)$, the probability that a random data vector will be in class k ,
- A_k , a mixing matrix for each class k that defines a plurality of basis functions,
- b_k , a bias vector of bias parameters for each of said basis functions in the mixing matrix for the class k , and
- 30 β_k , a vector of pdf parameters for each of said basis functions in the mixing matrix for the class k , each of said pdf parameters indicative of one of a plurality of generalized Gaussian pdfs; and
- in a main adaptation loop, for each data vector x_t from data index $t = 1$ to $t = T$, performing the steps of adapting the class parameters including the mixing matrices,

- bias vectors, and pdf parameters for each class from class index $k = 1$ to $k = K$;
 repeating the main adaptation loop a plurality of iterations while observing a
 learning rate at each subsequent iteration;
 after observing convergence of said learning rate, then assigning each data
 5 vector to one of said classes; and
 utilizing the source vectors for the assigned classes to separate the source
 signals in each of said classes.
17. The method of claim 16 wherein said step of adapting said each of said pdf parameters
 includes determining the maximum posteriori value for each of said pdf parameters.
- 10 18. The method of claim 16 wherein said main adaptation loop comprises:
 performing an initial calculation loop for each class, including calculating a
 source vector \mathbf{s}_k , a probability of the source vector $p(\mathbf{s}_k)$ and the likelihood of the data
 $p(\mathbf{x}_i | \theta_k, C_k)$; and
 performing a class probability loop that determines the class probability for
 15 each class $p(C_k | \mathbf{x}_i, \Theta)$.
19. The method of claim 16 further comprising the step of utilizing the adapted class
 parameters to estimate source data densities.
20. The method of claim 16 wherein said pdfs of said sources include at least one super-
 Gaussian pdf and at least one sub-Gaussian pdf, and said step of adapting the class parameters
 20 further includes adapting the pdf parameters for each source.
21. An apparatus that utilizes a generalized Gaussian mixture model to adapt class
 parameters and classify a plurality of data vectors having N elements using a plurality of
 generalized Gaussian probability density functions (pdfs) comprising:
 a data memory for storing a plurality of data vectors \mathbf{x}_t from data index $t = 1$ to
 25 $t = T$;
 a class parameter memory for storing initial parameters for each class, including
 K , the number of classes,
 $p(C_k)$, the probability that a random data vector will be in class k ,
 \mathbf{A}_k , a mixing matrix for each class k that defines a plurality of basis
 30 functions,
 \mathbf{b}_k , a bias vector of bias parameters for said basis functions in the mixing
 matrix for the class k , and
 β_k , a vector of pdf parameters for said basis functions in the mixing

- matrix for the class k , each of said pdf parameters indicative of one of a plurality of generalized Gaussian pdfs; and
calculation means, in a main adaptation loop, for adapting the class parameters for each data vector \mathbf{x}_t from data index $t = 1$ to $t = T$, comprising
- 5 means for adapting a mixing matrix \mathbf{A}_k for each class k , including means for associating each basis function in said mixing matrix with a pdf parameter, and
- means for adapting a bias vector \mathbf{b}_k for each class from class index $k = 1$ to $k = K$;
- 10 means for adapting a pdf vector β_k for each class from class index $k = 1$ to $k = K$;
- learning means for repeating the main adaptation loop a plurality of iterations while observing a learning rate at each subsequent iteration; and
- a control system for observing convergence of said learning rate, and assigning
- 15 each data vector to one of said classes.
22. The apparatus of claim 21 wherein said step of adapting each of said pdf vectors includes determining the maximum posteriori value for each of the pdf parameters within said pdf vectors.
23. The apparatus of claim 21 further comprising the step of utilizing the source vectors for
- 20 the assigned classes to separate source signals in each of said classes.
24. The apparatus of claim 21 further comprising means for adapting the number of classes after at least one iteration of said main adaptation loop.
25. The apparatus of claim 21 further comprising a system for supplying additional data vectors and utilizing the adapted class parameters to classify each of said additional data
- 25 vectors into one of said k classes.
26. The apparatus of claim 25 further comprising an encoding system that utilizes the adapted class parameters to encode source data.
27. The apparatus of claim 26 further comprising a classification system, responsive to said adapted class parameters, that classifies said encoded source data.
- 30 28. The apparatus of claim 26 wherein said source data comprises image data indicative of an image, and further comprising a system, responsive to said adapted class parameters, that

classifies said encoded image data and reconstructs said image.

29. The apparatus of claim 26 wherein said source data comprises speech data, and further comprising a system, responsive to said adapted class parameters, that classifies said encoded speech data and reconstructs said speech.

5 30. The classification apparatus of claim 21 further comprising a computer, wherein said data memory, said class parameter memory, said calculation means, said learning means, and said control system are implemented in said computer.

31. A classification apparatus for classifying a plurality of data vectors using known class parameters including a predetermined mixing matrix for each class that includes a plurality of
10 basis functions and a pdf parameter for each basis function that is indicative of one of a plurality of generalized Gaussian probability density functions, comprising:

a data memory for storing a plurality of data vectors \mathbf{x}_t from data index $t = 1$ to $t = T$;

a class parameter memory for storing class parameters including
15 K , the number of classes,
 $p(C_k)$, the probability that a random data vector will be in class k ,
 \mathbf{A}_k , a mixing matrix for each class k that defines a plurality of basis functions,
 \mathbf{b}_k , a bias vector of bias parameters for each of said basis functions in the
20 mixing matrix for the class k , and
 β_k , a vector of pdf parameters for each of said basis functions in the mixing matrix for the class k , each of said pdf parameters indicative of one of a plurality of generalized Gaussian pdfs;
a calculation system for calculating the class probability for each data vector
25 responsive to class parameters for each class; and
an assignment system that assigns each data vector to one of said classes responsive to said calculated class probability.

32. The classification apparatus of claim 31 wherein calculation system comprises:
means for performing an initial calculation loop for each class, including
30 calculating a source vector \mathbf{s}_k responsive to said class parameters, a probability of the source vector $p(\mathbf{s}_k)$ and the likelihood of the data $p(\mathbf{x}_t | \theta_k, C_k)$; and
means for performing a class probability loop that determines the class probability for each class $p(C_k | \mathbf{x}_t, \Theta)$ for each data vector.

33. The classification apparatus of claim 31 wherein said data memory includes data vectors from an image, and further comprising an image reconstruction system for reconstructing said image.
34. The classification apparatus of claim 31 wherein said data memory includes data
5 vectors from at least one speech source.
35. The classification apparatus of claim 31 further comprising a medical data classification system for receiving and classifying medical data.
36. The classification apparatus of claim 31 further comprising a computer, wherein said
10 data memory, said class parameter memory, said calculation system, and said assignment system are implemented in said computer.

1/20

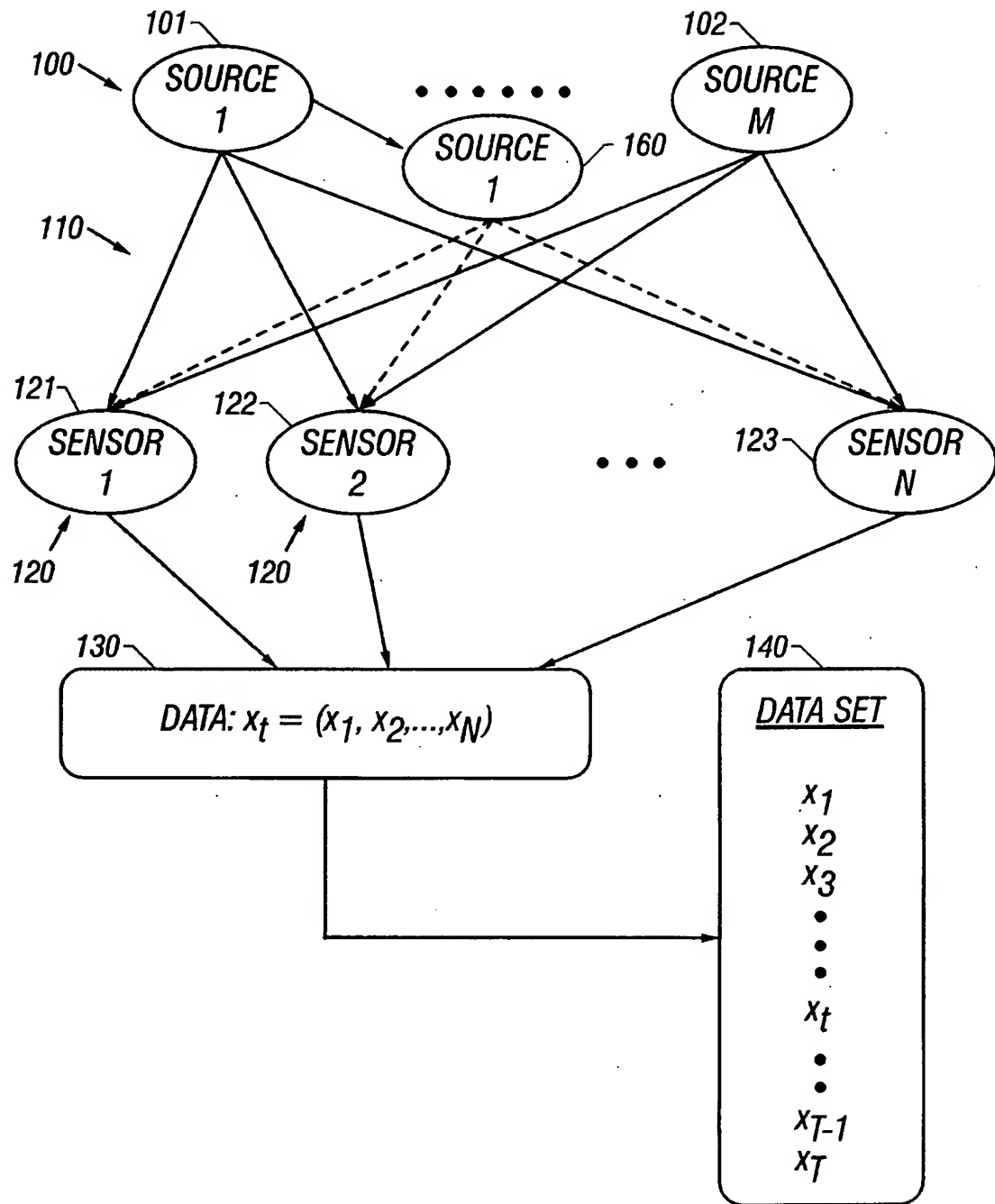


FIG. 1

2/20

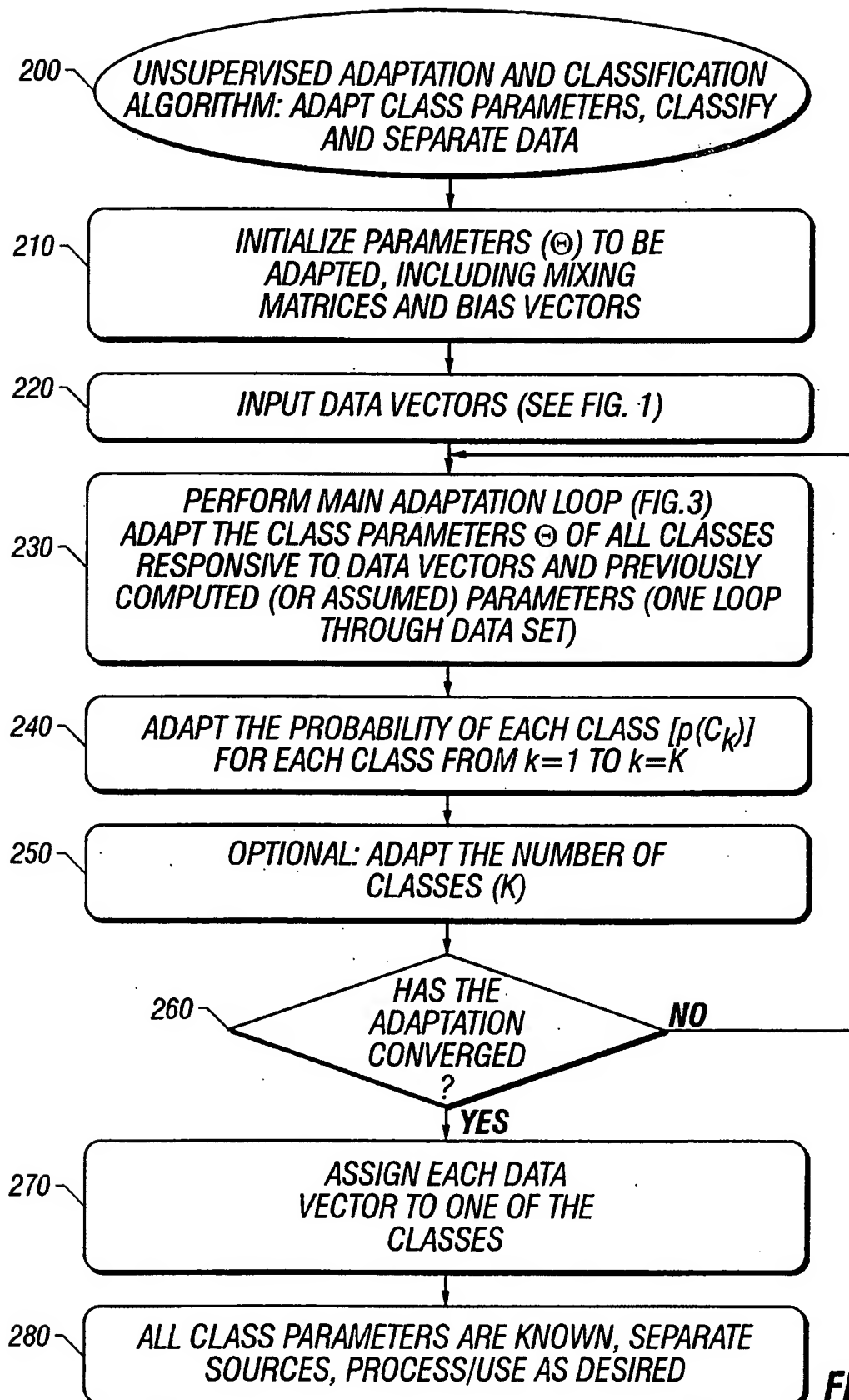


FIG. 2

3/20

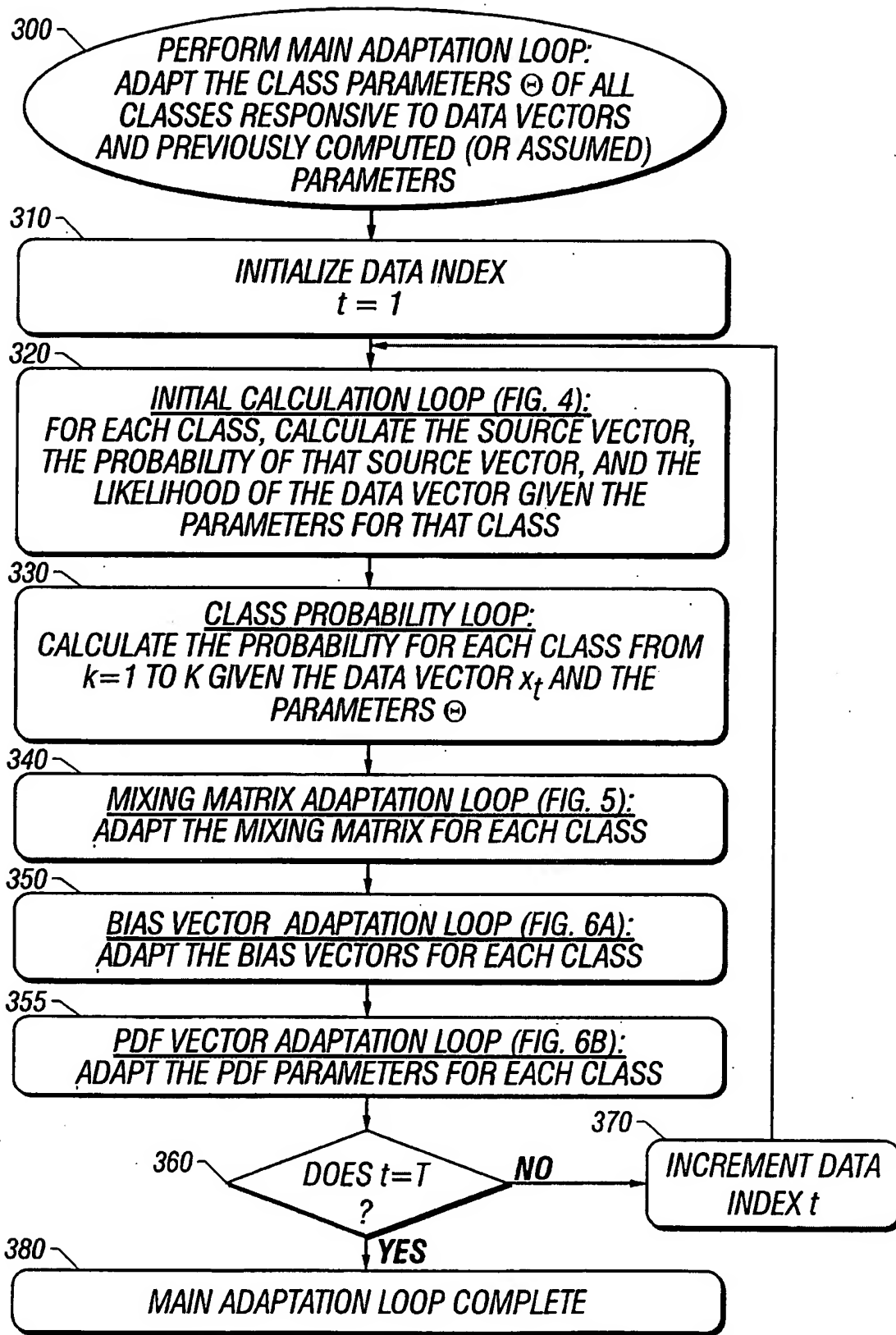


FIG. 3

4/20

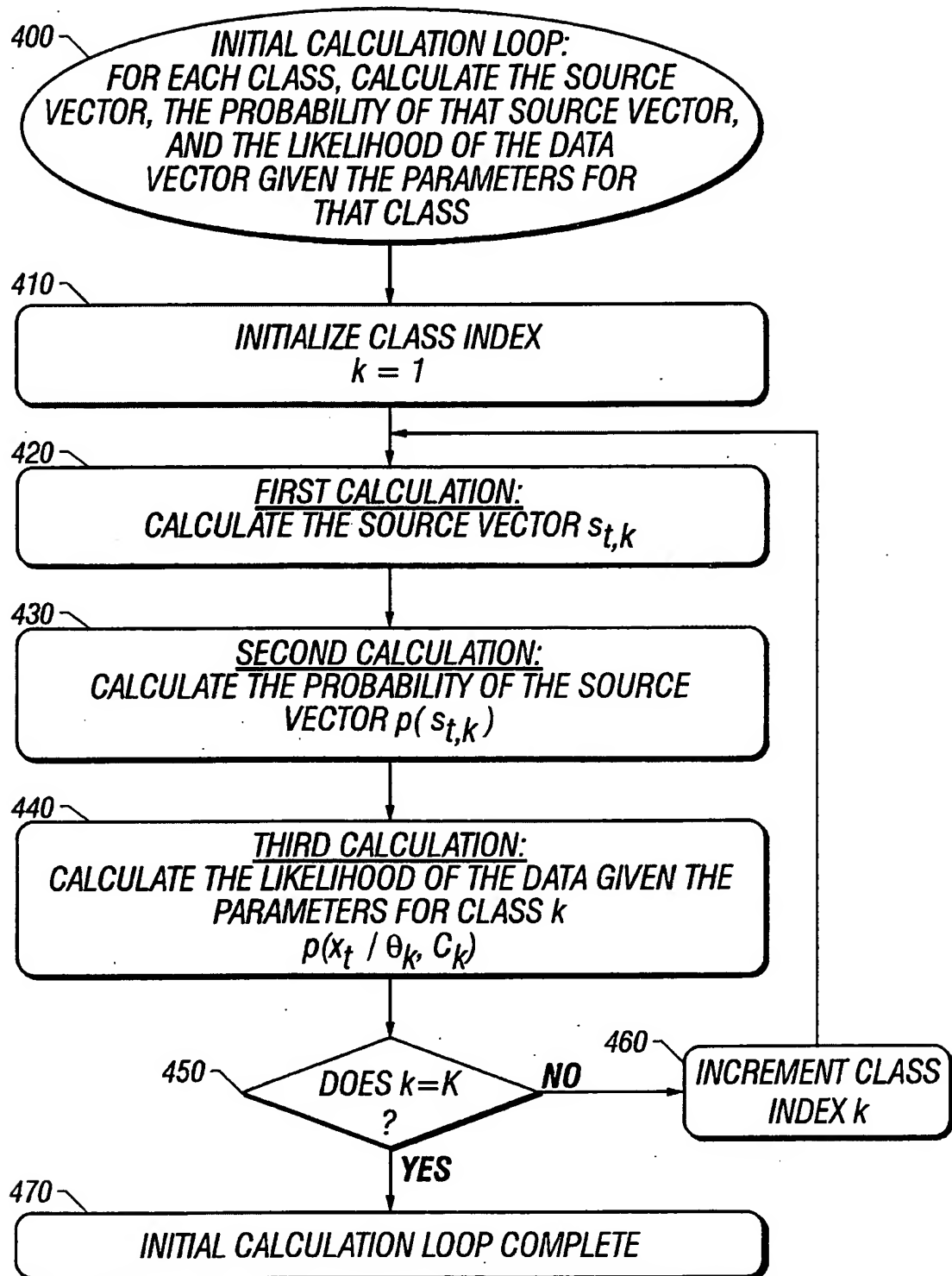


FIG. 4

5/20

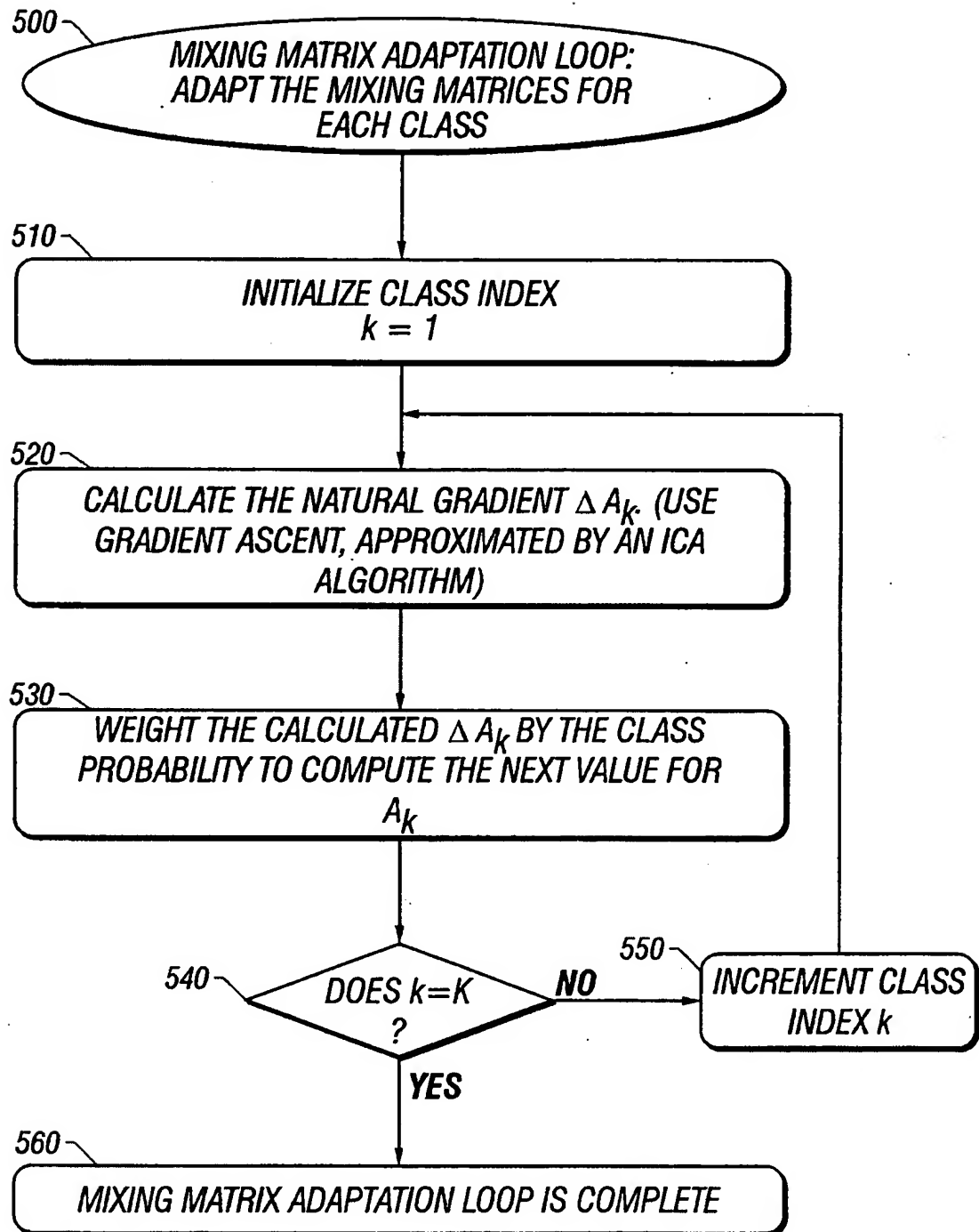


FIG. 5

6/20

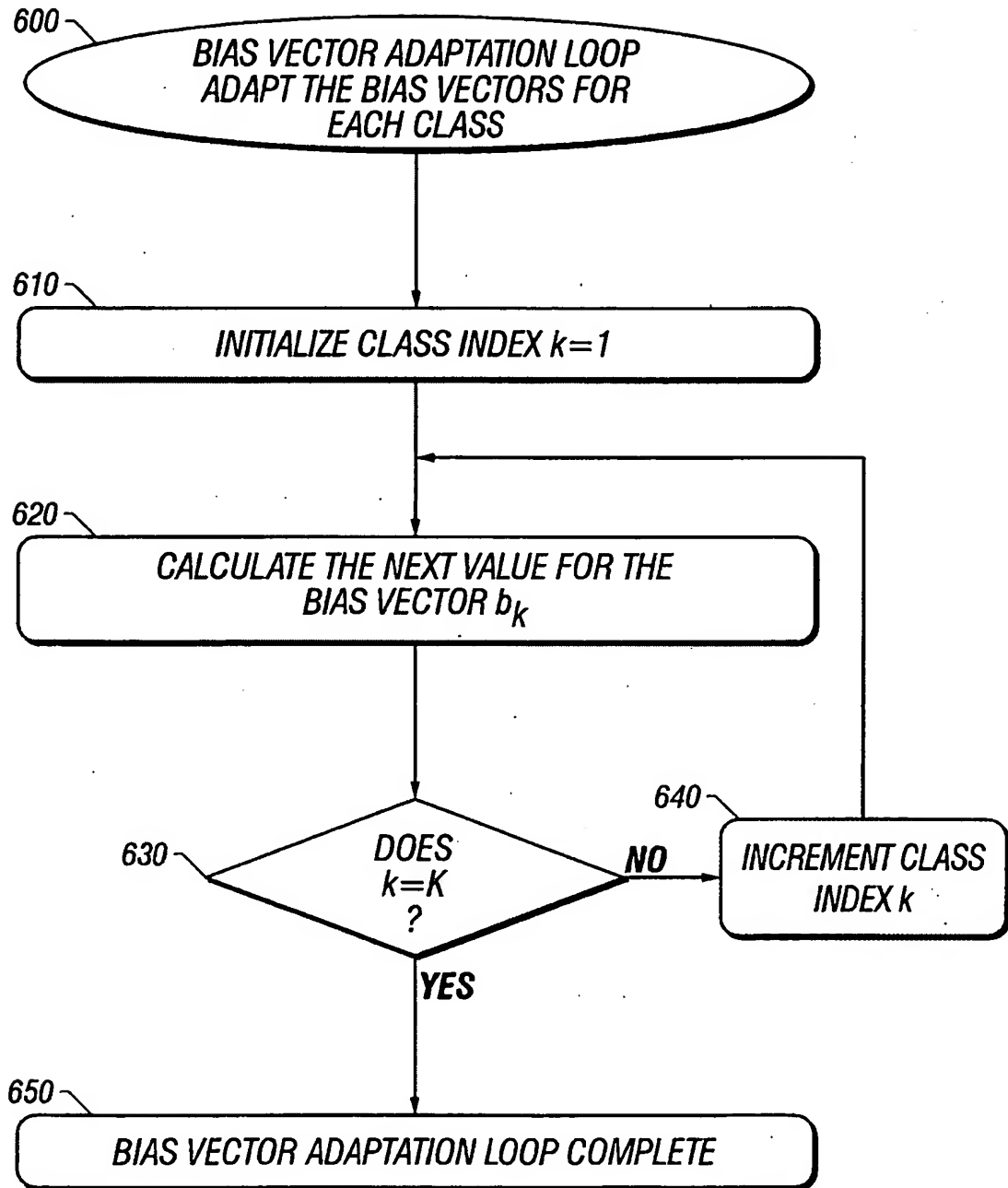


FIG. 6A

7/20

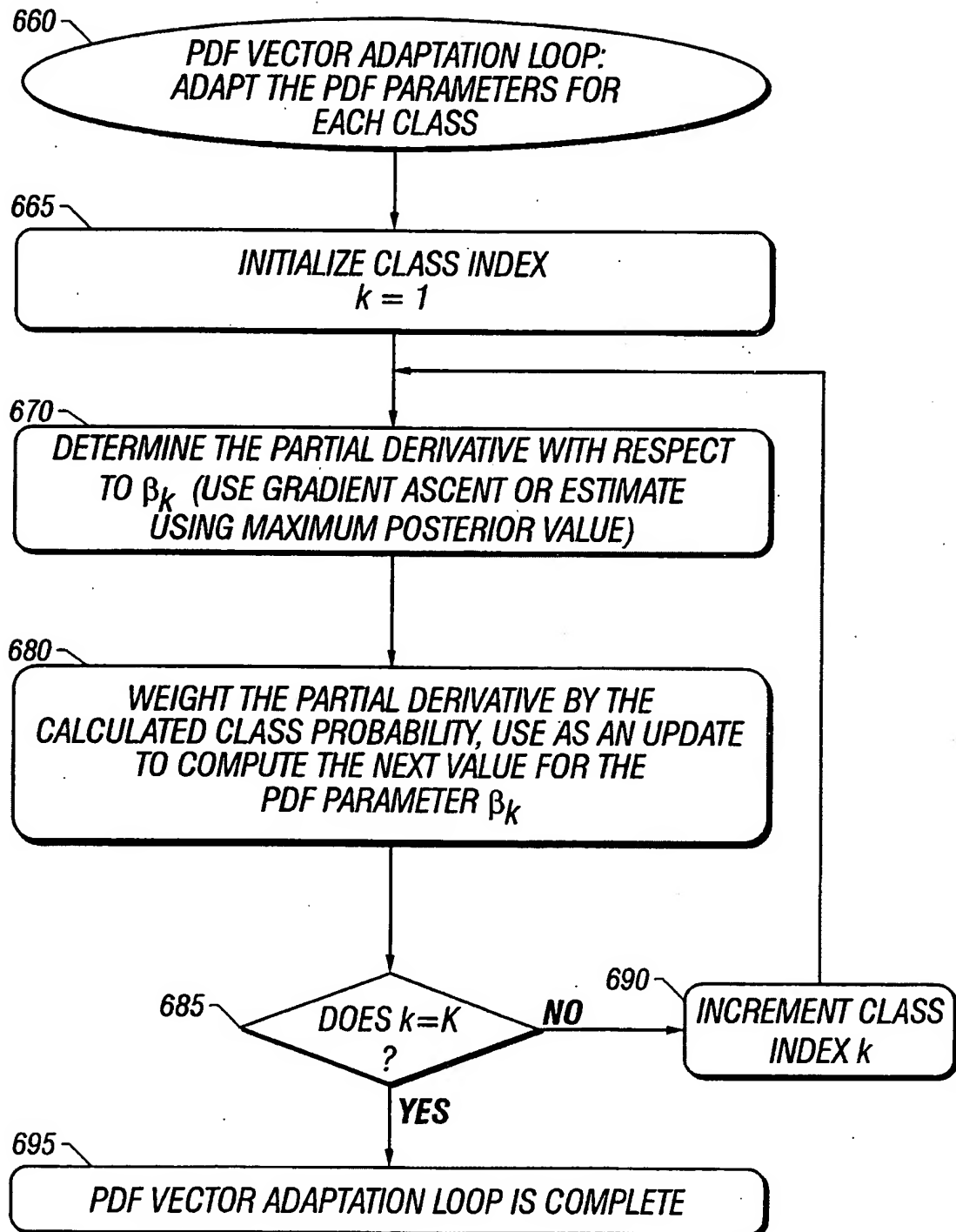


FIG. 6B

8/20

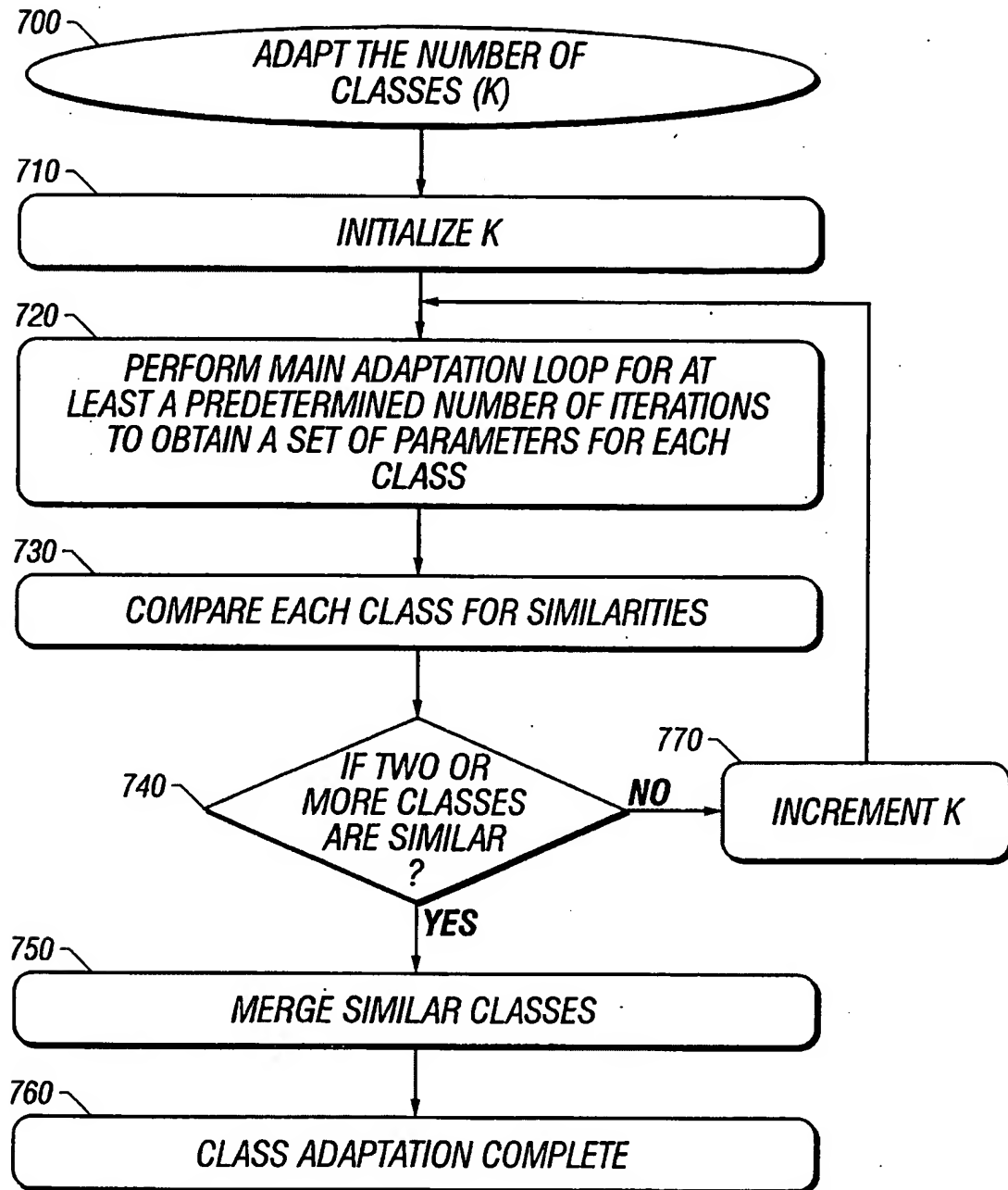


FIG. 7

9/20

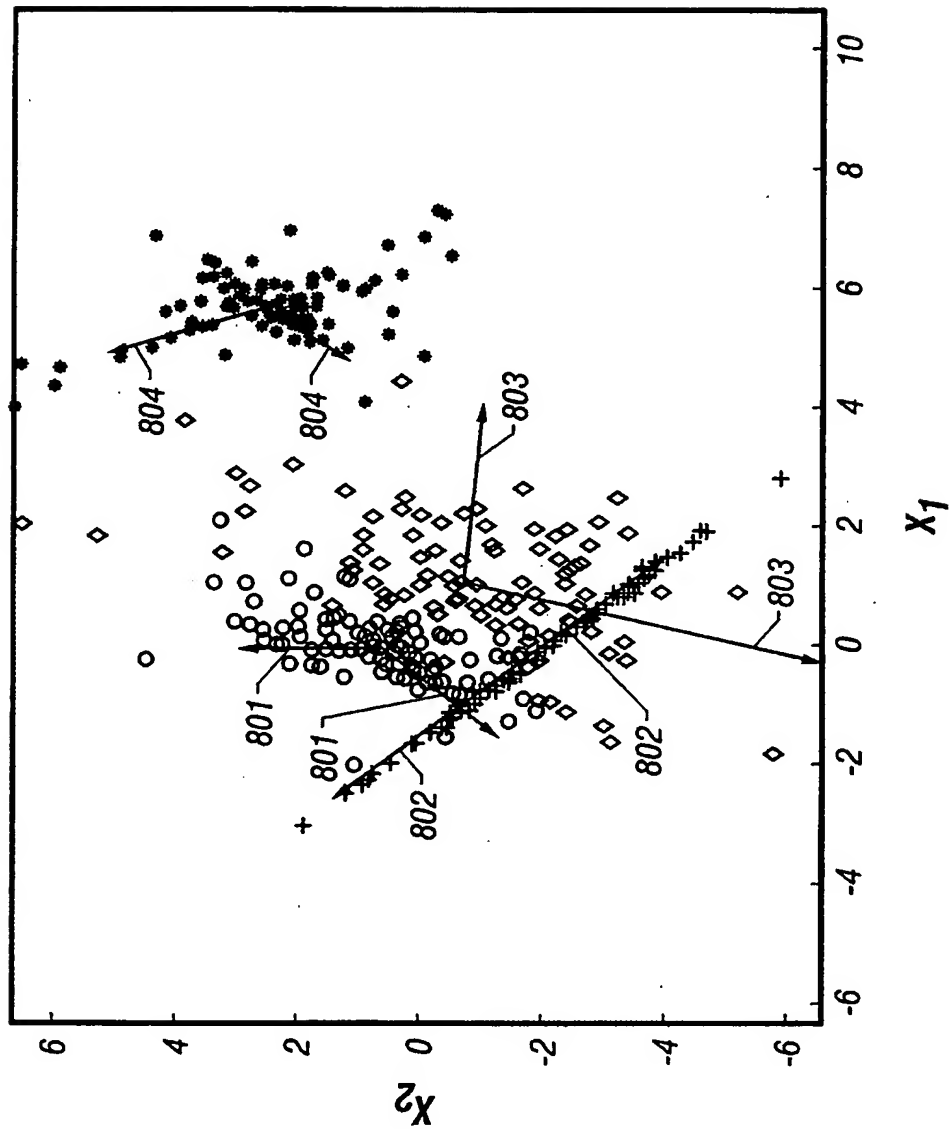
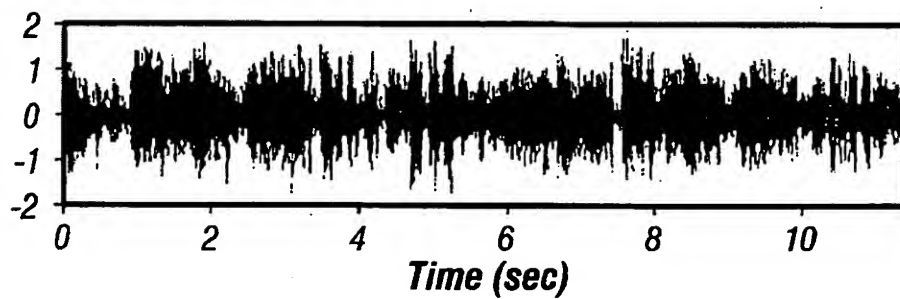
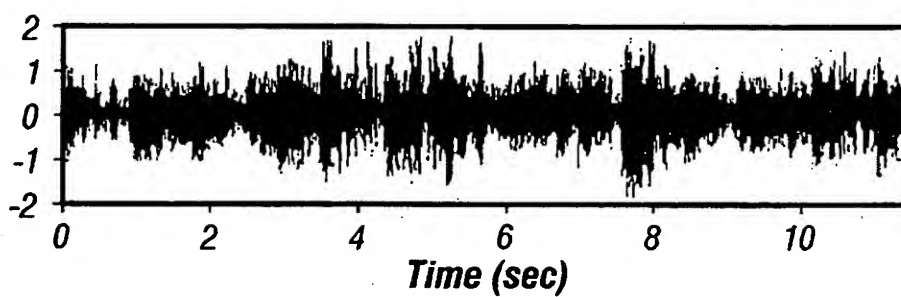
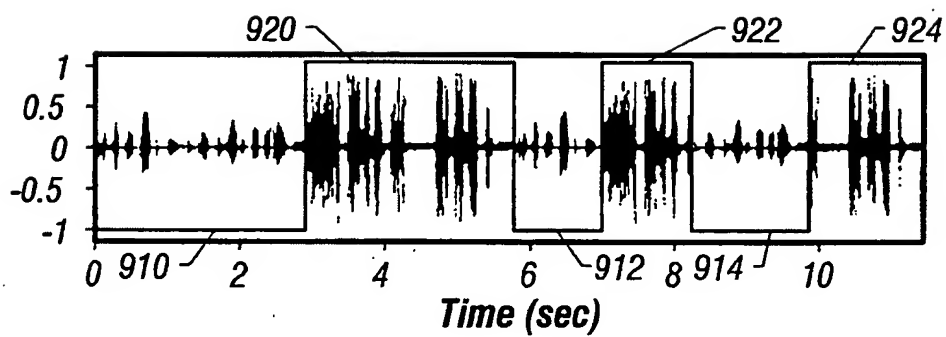
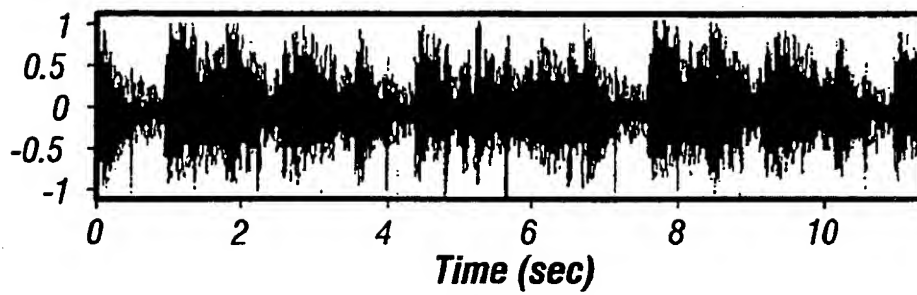
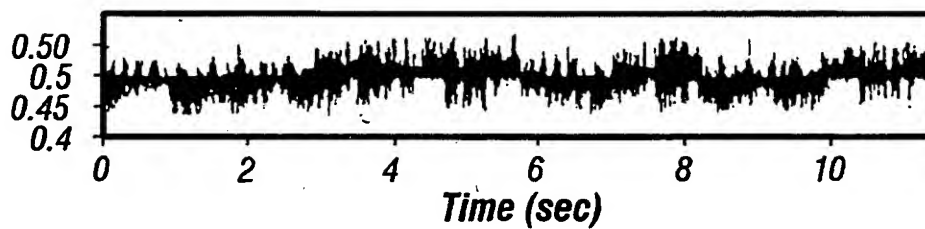
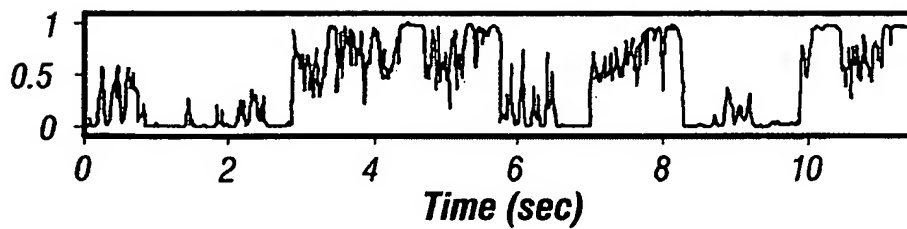
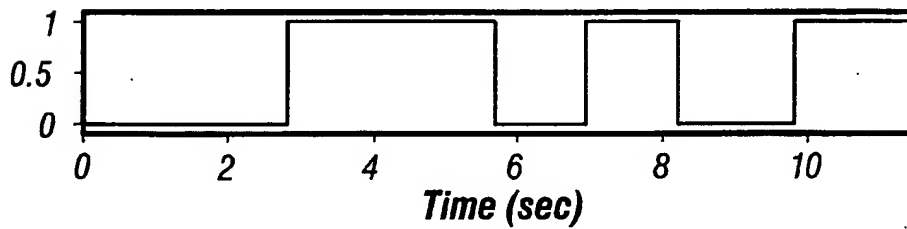


FIG. 8

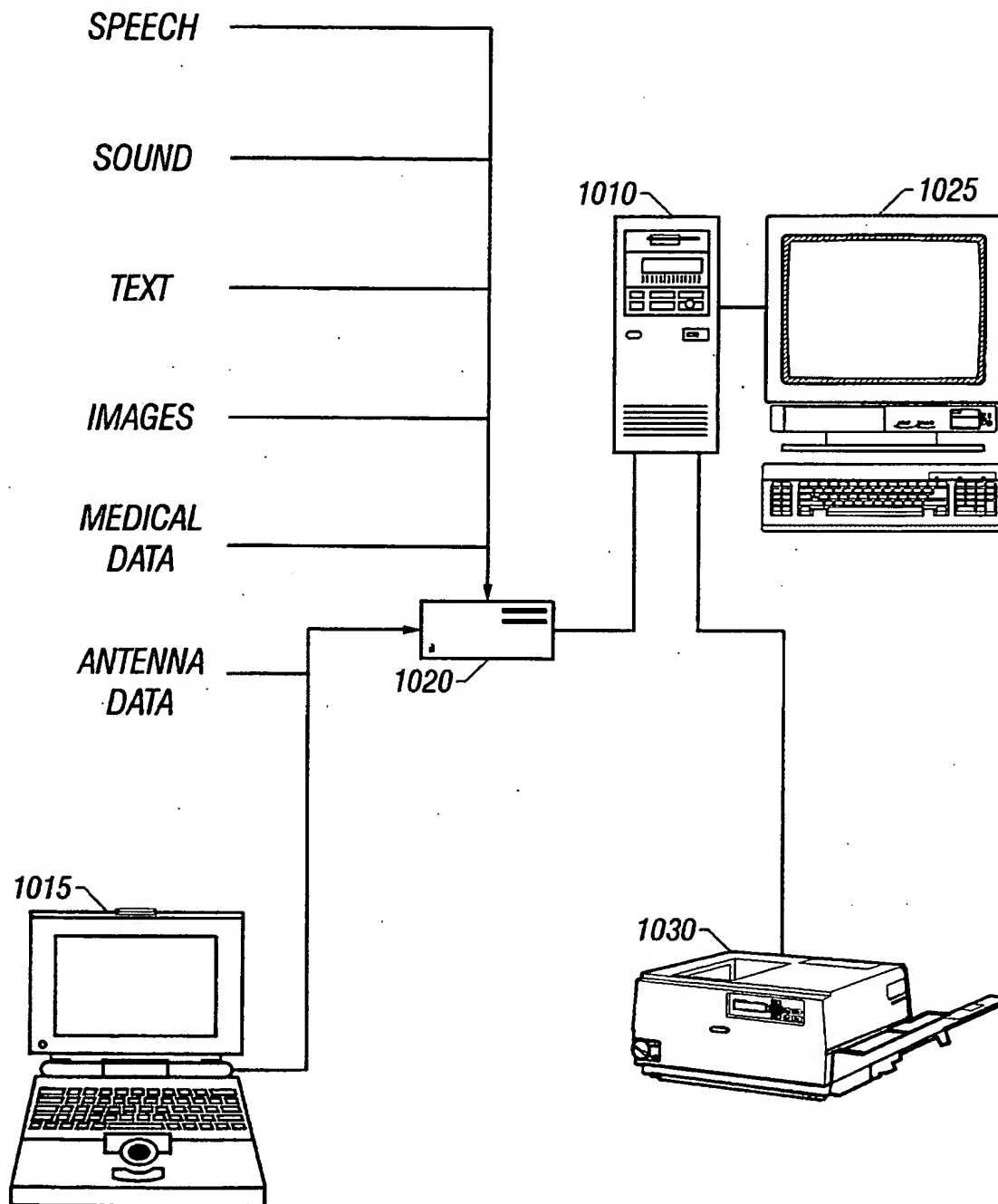
10/20

**FIG. 9A****FIG. 9B****FIG. 9C**

11/20

**FIG. 9D****FIG. 9E****FIG. 9F****FIG. 9G**

12/20

**FIG. 10**

13/20

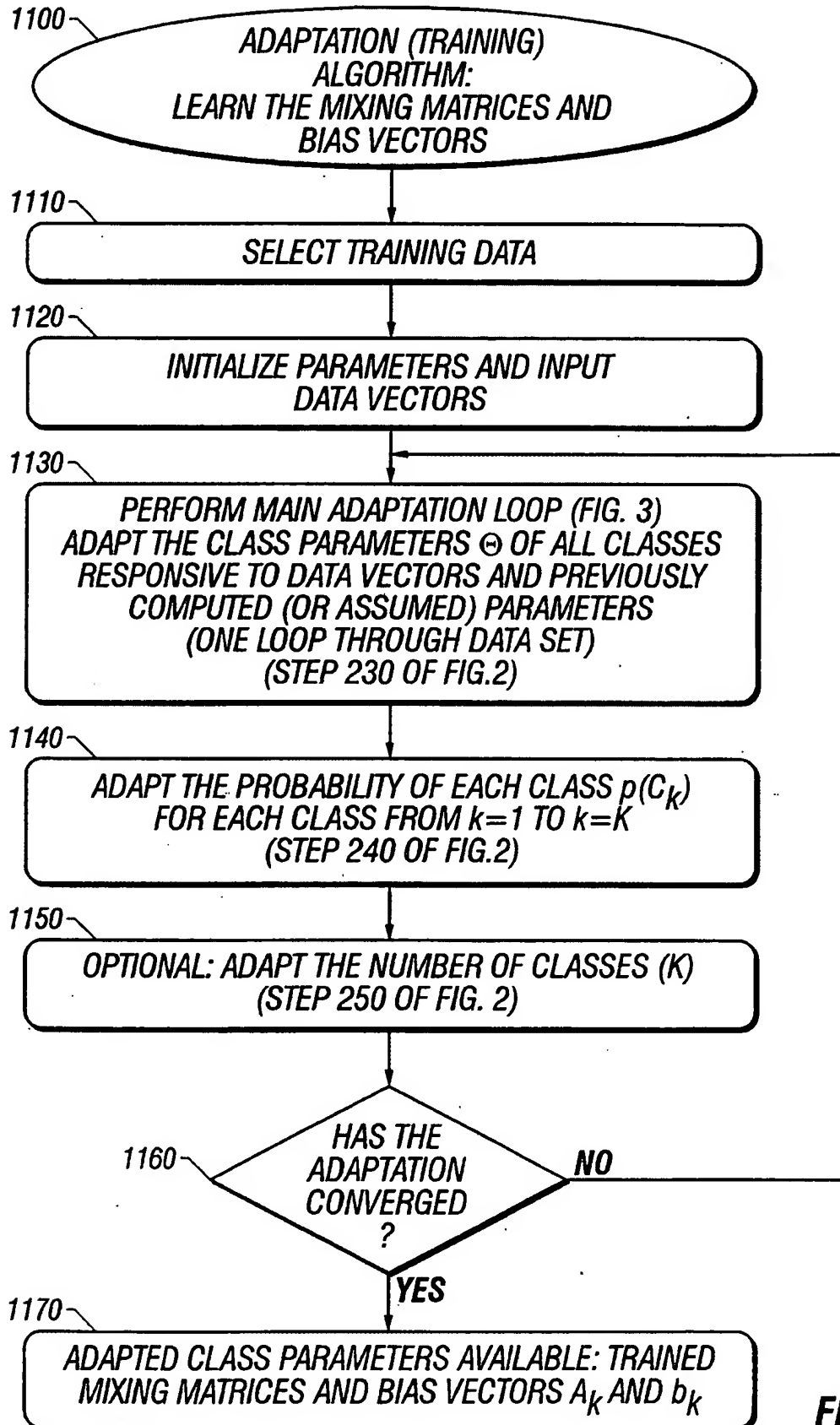


FIG. 11

14/20

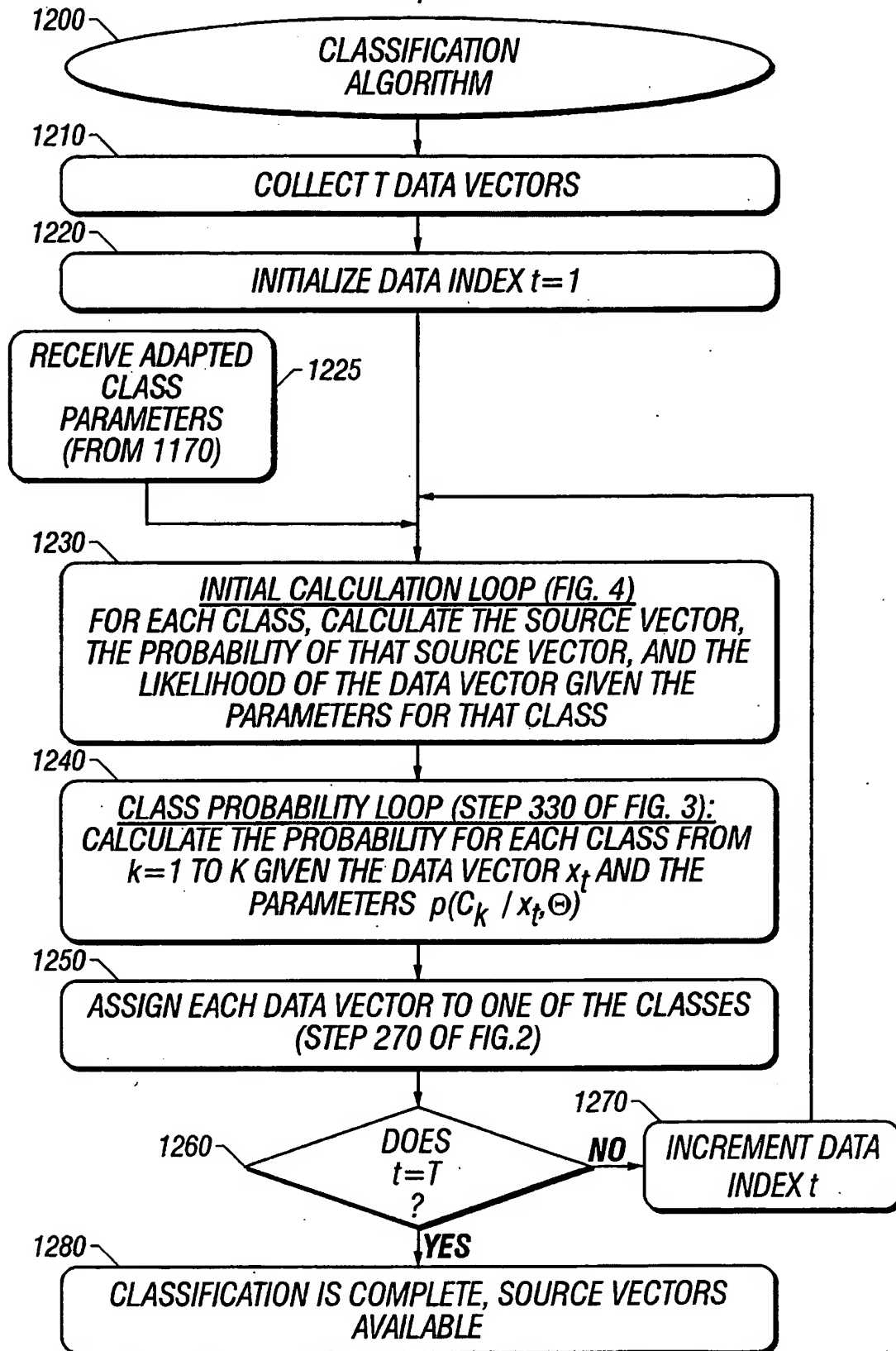
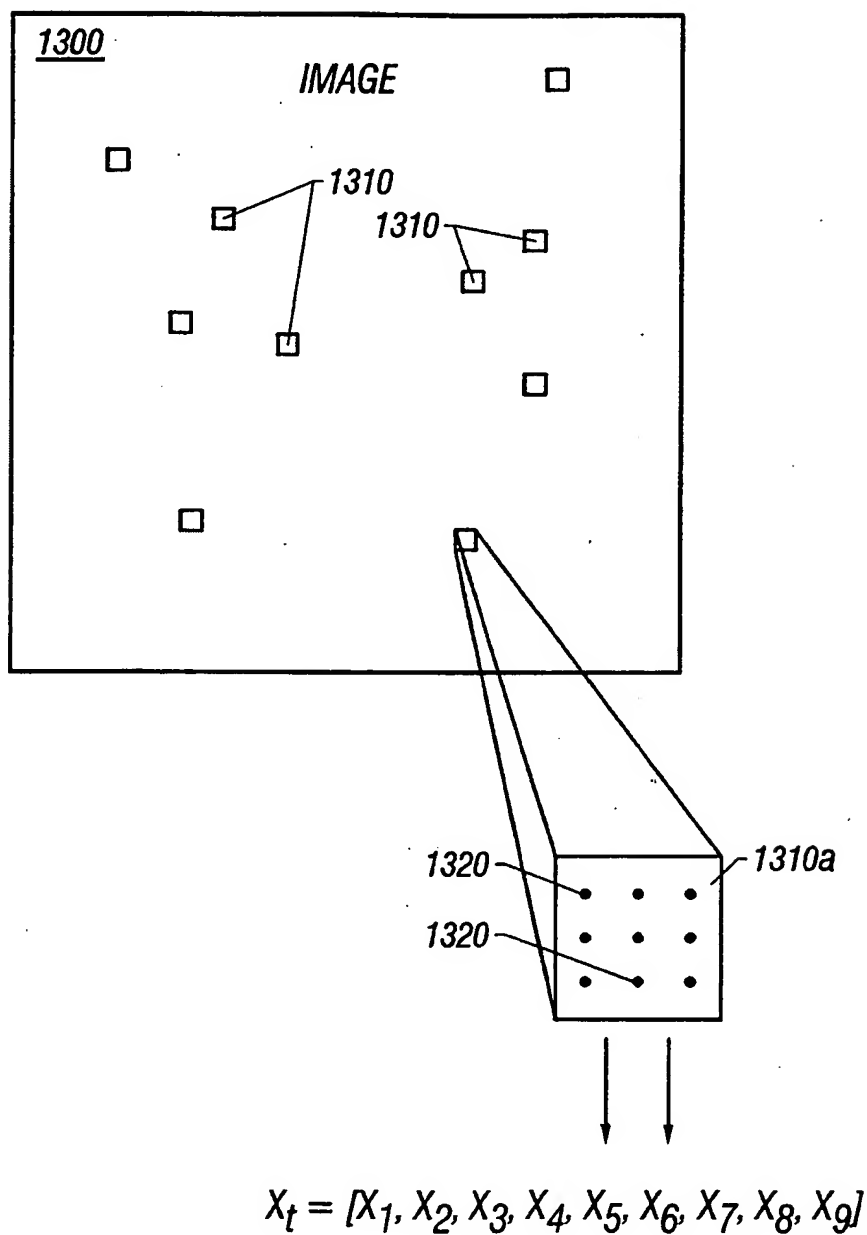


FIG. 12

15/20

**FIG. 13**

16/20

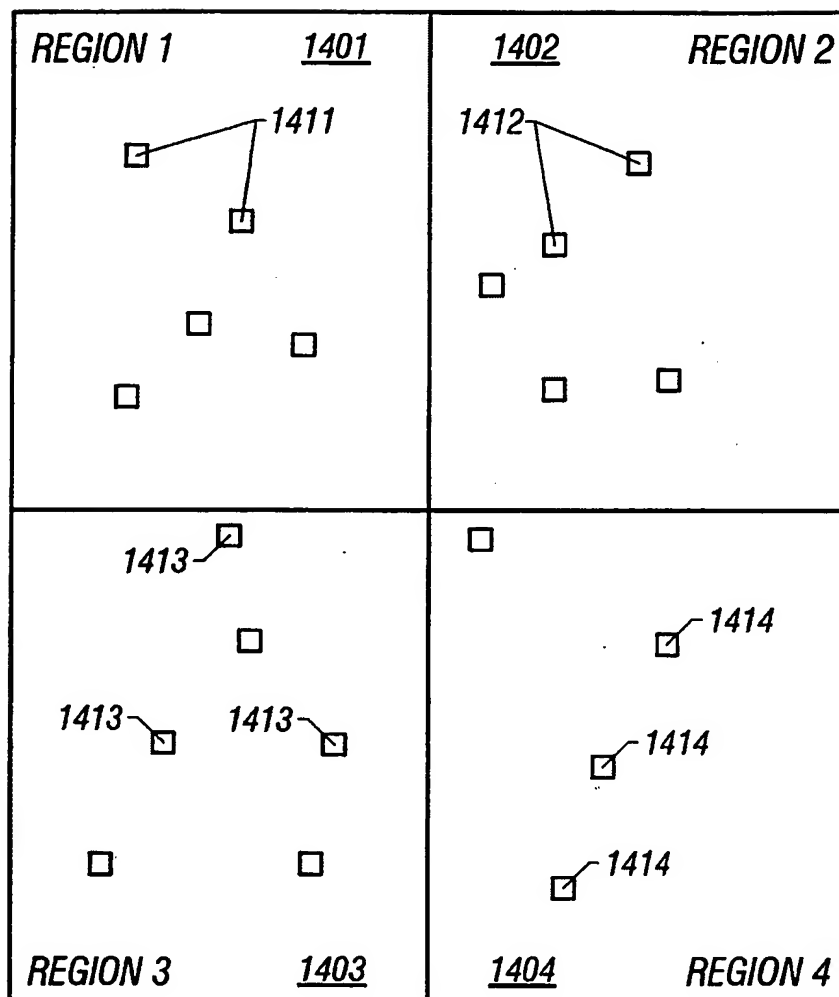


FIG. 14

17/20

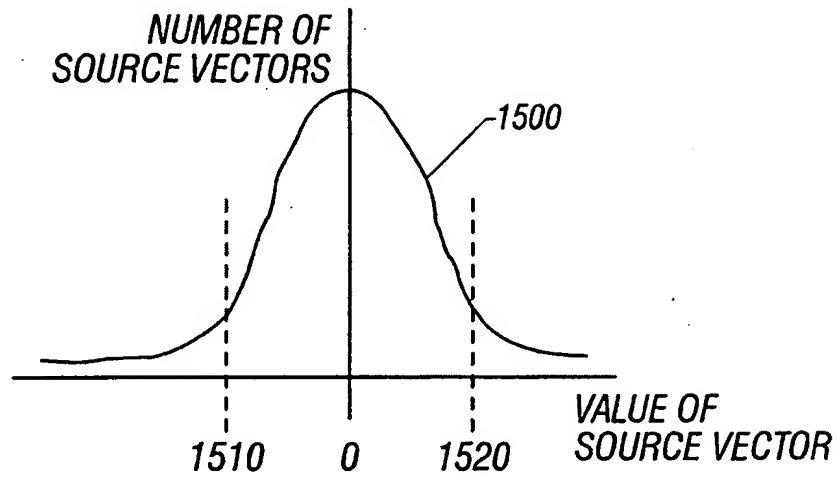


FIG. 15

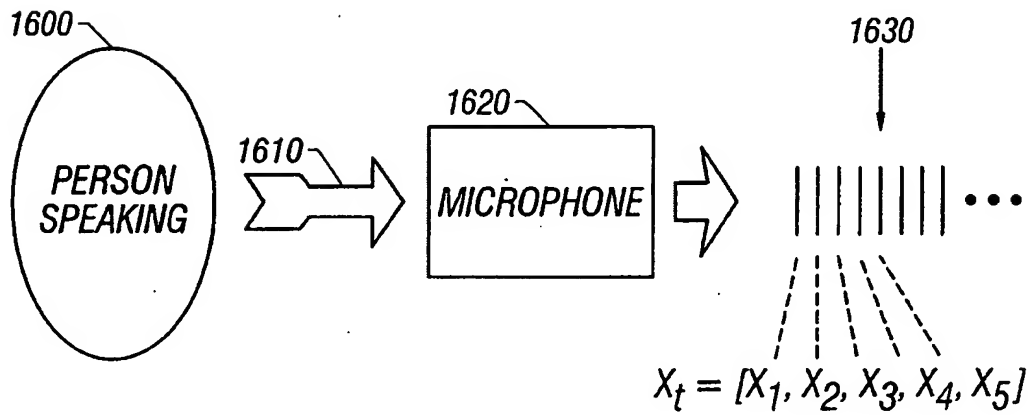


FIG. 16

18/20

$\beta = -0.75 \gamma_2 = -1.08$

$\beta = -0.25 \gamma_2 = -0.45$

$\beta = +0.00 \text{ (NORMAL)} \gamma_2 = +0.00$

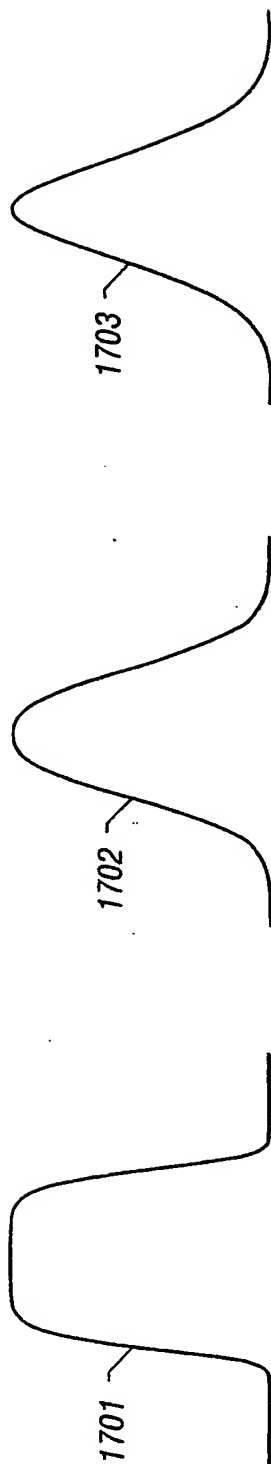


FIG. 17A

FIG. 17B

FIG. 17C

$\beta = +0.50 \text{ (ICA TANH)} \gamma_2 = +1.21$ $\beta = +1.00 \text{ (LAPLACIAN)} \gamma_2 = +3.00$ $\beta = +2.00 \gamma_2 = +9.26$

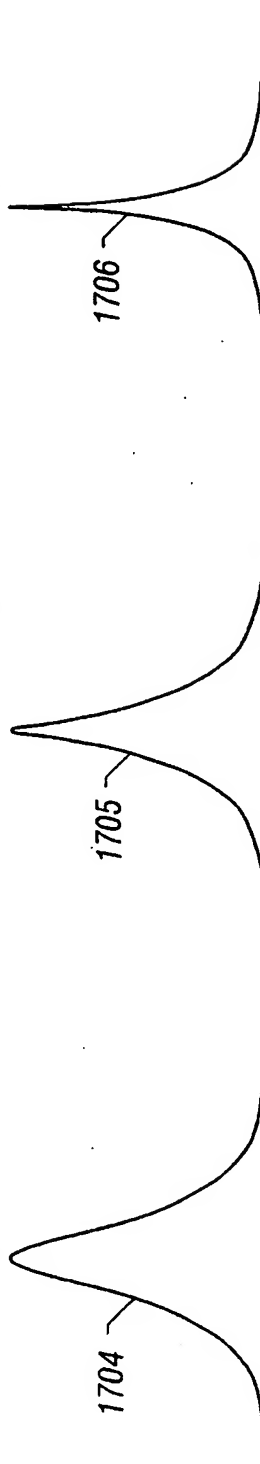


FIG. 17D

FIG. 17E

FIG. 17F

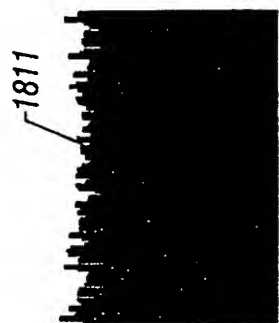


FIG. 18B

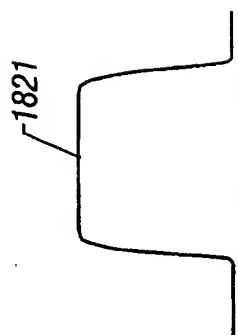


FIG. 18D

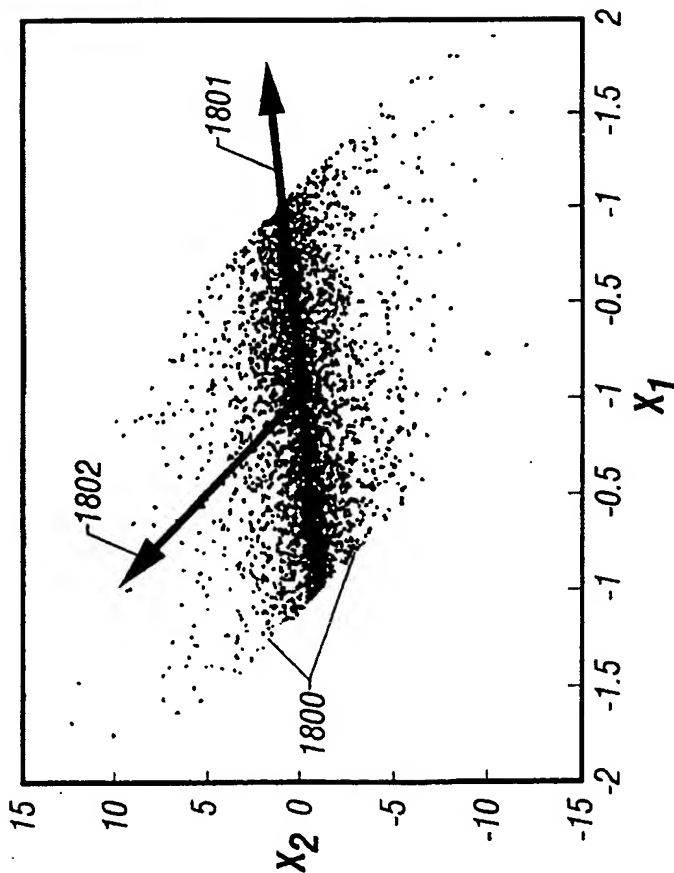


FIG. 18A

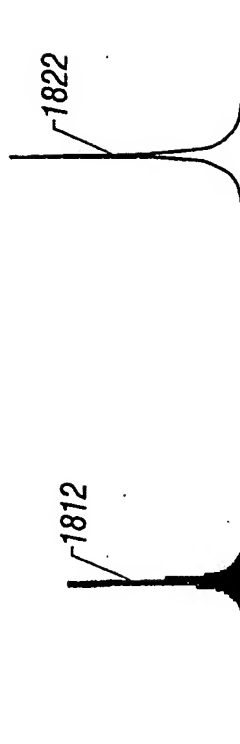


FIG. 18C

FIG. 18E

20/20

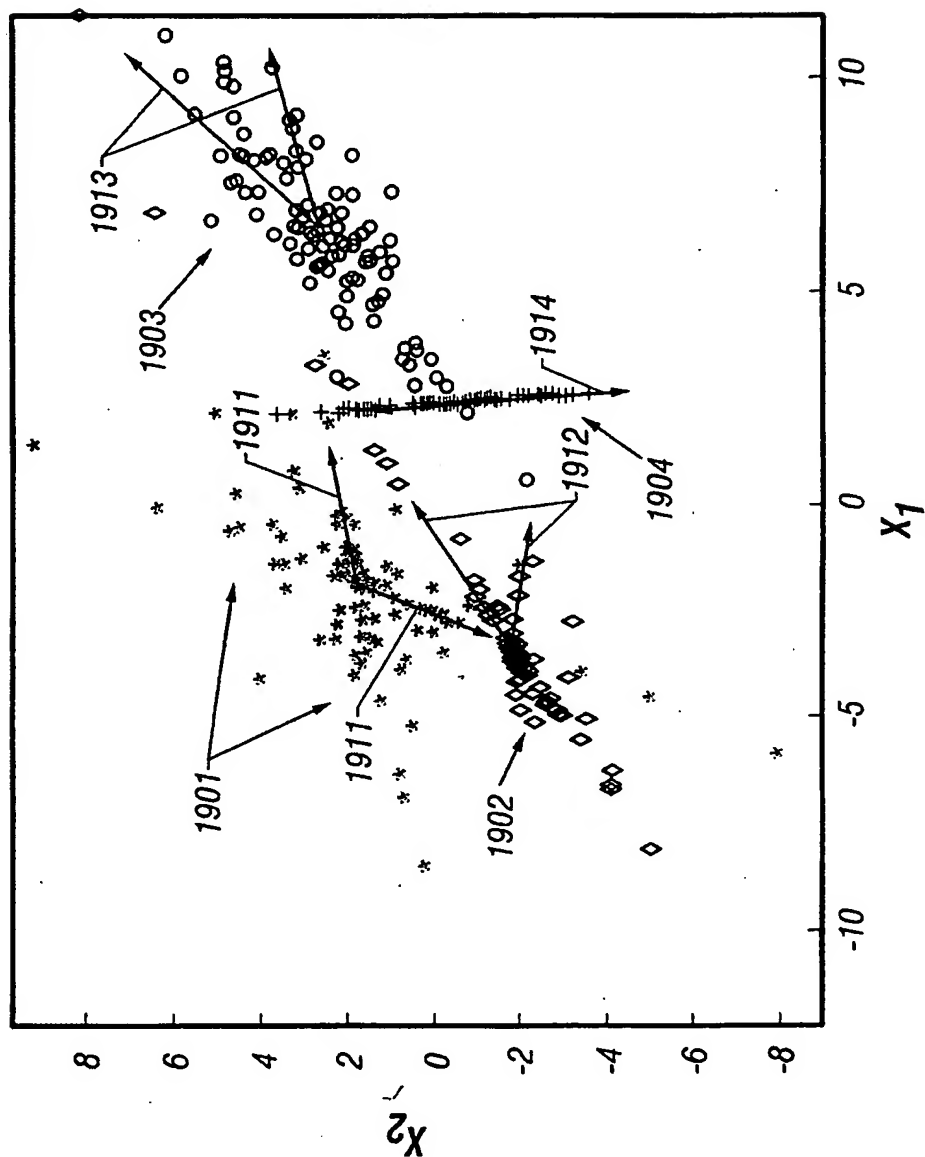


FIG. 19

INTERNATIONAL SEARCH REPORT

International application No.

PCT/US00/28453

A. CLASSIFICATION OF SUBJECT MATTER

IPC(7) :G06N 3/02

US CL : 706/15

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 706/15

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EAST, INTERNET, IEEE

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A,P	US 5,999,902 A (SCAHILL et al) 07 December 1999, col. 2, lin. 1-67.	1-36
A	US 5,706,402 A (BELL) 06 January 1998, col. 18, lin. 30-67.	1-36
A	US 5,706,391 A (YAMADA et al) 06 January 1998, Fig. 9.	1-36
A	US 5,778,342 A (ERELL et al) 07 July 1998, Fig.1.	1-36
A	US 5,625,749 A (GOLDENTHAL et al) 29 April 1997, col. 21, lin. 29-52.	1-36
A	US 5,724,487 A (STREIT) 03 March 1998, col. 14, lin. 1-67.	1-36

☒ Further documents are listed in the continuation of Box C.☐ See patent family annex.

* Special categories of cited documents:	"T" later documents published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"A" document defining the general state of the art which is not considered to be of particular relevance	"X" documents of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"E" earlier document published on or after the international filing date	"Y" documents of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"&" document member of the same patent family
"O" document referring to an oral disclosure, use, exhibition or other means	
"P" document published prior to the international filing date but later than the priority date claimed	

Date of the actual completion of the international search

24 NOVEMBER 2000

Date of mailing of the international search report

31 JAN 2001

Name and mailing address of the ISA/US
Commissioner of Patents and Trademarks
Box PCT
Washington, D.C. 20231

Facsimile No. (703) 305-3230

Authorized officer

WILBERT L. STARKS JR.

Telephone No. (703) 308-9700

Form PCT/ISA/210 (second sheet) (July 1998)*

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US00/28453

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 5,790,758 A (STREIT) 04 August 1998, col. 10, lin. 9-59	1-36
A	US 5,933,806 A (BEYERLEIN et al) 03 August 1999, col. 9, lin. 1-65.	1-36

Form PCT/ISA/210 (continuation of second sheet) (July 1998)*